

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO MULTIDISCIPLINAR

MICHEL DIAS DE ARRUDA
RAFAEL OLIVEIRA COSTA

**Uma ferramenta de análise de padrões
em *hashtags* no Twitter**

Prof. Carlos Eduardo R. de Mello, Ph.D.
Orientador

Nova Iguaçu, Fevereiro de 2016

Uma ferramenta de análise de padrões em *hashtags* no Twitter

Michel Dias de Arruda

Rafael Oliveira Costa

Projeto Final de Curso submetido ao Departamento de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Apresentado por:

Michel Dias de Arruda

Rafael Oliveira Costa

Aprovado por:

Prof. Carlos Eduardo R. de Mello, Ph.D.

Prof. Filipe Braida do Carmo, M.Sc.

Prof. Leandro Guimarães M. Alvim, D.Sc.

NOVA IGUAÇU, RJ - BRASIL

Fevereiro de 2016

Agradecimentos

Michel Dias de Arruda

Agradeço primeiramente a Deus, por sempre me fortalecer e me guiar em todos meus objetivos.

Também a Laercio Arruda e Rosane Arruda, meus pais, por terem me gerado, educado, amado e auxiliado em toda vida.

A minha irmã Letícia Arruda, por sempre me incentivar a prosseguir.

A minha companheira e amada Airine Carmo, por estar ao meu lado durante todo o curso, me incentivando e auxiliando.

Aos meus amigos de curso, principalmente aos que se intitulam "Morcegos", que fizeram o curso ser mais alegre e divertido, que dividiram muitos sofrimentos, cansaços e cafés.

Ao Professor Carlos, meu orientador, pela oportunidade e paciência na realização deste trabalho, que espero fazer jus a confiança dedicada a mim.

Aos professores Filipe Braida do Carmo e Leandro Guimarães M. Alvim, por aceitarem fazer parte da banca examinadora, abrindo mão de seus compromissos e sacrificando o tempo de suas agendas.

A todos professores que me acompanharam durante a graduação.

E a todos aqueles que me auxiliaram de alguma forma durante o curso.

Rafael Oliveira Costa

Agradeço primeiramente a meus pais, Edio Alves e Rosicleide da Silva, por terem me dado todo amor e educação durante minha vida.

A minha namorada e companheira Carla Eduarda, por ter estado ao meu lado e tido paciência durante todos esses anos.

Aos meus grande amigos que fiz no curso: Michel Arruda, Egberto Caetano, Ricardo Luiz, Luis Paulo, Miguel Mendes, Válber Laux, Ygor Canalli, Alexsander Melo, Pedro Aguiar, Raul Sena, Kleyton Pontes, Hugo Rebelo, Leonardo Duarte, Luiz Fernando, Victor Pedro, Renan Sies, Juliane Marinho, Larissa Ferrarez, Suzane Lima, kesia Braga, Jéssica Aparecida, Liliane Neves, Jamile Santos e Karine Gomes.

Ao meu orientador, Carlos Eduardo Mello, pela grande oportunidade e apoio na realização deste trabalho.

Aos professores Filipe Braida e Leandro Alvim, por aceitarem fazer parte da banca examinadora e fazer parte deste grande momento pra mim.

A todos os professores e técnicos que fizeram parte da minha graduação. E a todos que me ajudaram direta ou indiretamente durante todos esses anos de curso.

RESUMO

Uma ferramenta de análise de padrões em *hashtags* no Twitter

Michel Dias de Arruda e Rafael Oliveira Costa

Fevereiro/2016

Orientador: Carlos Eduardo R. de Mello, Ph.D.

Neste trabalho, uma solução de mineração de dados é desenvolvida cujo objetivo consiste na descoberta de padrões em *hashtags* a partir de dados do Twitter. Inicialmente, pode ser vista uma comparação entre algumas ferramentas similares encontradas na Internet, onde uma carência por ferramentas de mineração de dados em *hashtags* pode ser notada. Tal carência se torna alarmante devido à grande quantidade de dados oriundos de redes sociais, a qual inviabiliza a identificação de padrões por um processo de análise manual, e então, tornando-se necessária a criação desta solução.

Para tal, etapas de processamento que consistem na execução sequencial de workflows desenvolvidos foram utilizadas, que possibilitam a extração, tratamento, seleção, processamento e exibição dos padrões adquiridos. Por fim, a ferramenta foi aplicada a um estudo de caso, onde as hashtags *#foraDilma* e *#foraCunha* foram utilizadas como alvo de busca, apresentando resultados satisfatórios. A partir dos resultados, foi possível visualizar a forte ligação entre a presidente Dilma Rousseff e o seu partido, o PT, além de manifestações de repúdio à presidente. Além disso, pode ser visto que grande parte das manifestações contrárias a Eduardo Cunha ocorreram em virtude do pedido de impeachment da presidente Dilma Rousseff. Tais resultados reforçam a importância da mineração de dados no Twitter e a eficiência da ferramenta criada.

ABSTRACT

Uma ferramenta de análise de padrões em *hashtags* no Twitter

Michel Dias de Arruda and Rafael Oliveira Costa

Fevereiro/2016

Advisor: Carlos Eduardo R. de Mello, Ph.D.

In this work, a data mining solution is developed whose objective consists in the discovery of patterns in hashtags from Twitter data. Firstly, a comparison between some similar tools found on the Internet can be seen, where a shortage of data mining tools in hashtags can be noticed. Such shortage becomes alarming given the large amount of data from social networks, which makes the identification of patterns by a manual analysis process impracticable, and thus, making the creation of this solution necessary.

*To this end, processing steps involving the sequential execution of workflows were developed allowing the extraction, treatment, selection, processing and displaying of the acquired patterns. Finally, the tool was applied to a case study, where the hashtags *foraDilma* and *foraCunha* were used as search targets, presenting satisfactory results. From the results, it was possible to see the strong link between the President Dilma Rousseff and her party, the PT, as well as manifestations of repudiation to the President. Furthermore, it was possible to see that most of the protests against Eduardo Cunha occurred as a result of the request for the impeachment of the president Dilma Rousseff. Such results reinforce the importance of data mining on Twitter and the efficiency of the tool created.*

Lista de Figuras

Figura 2.1: Tela principal da ferramenta Hashtagfy	5
Figura 2.2: Dez maiores hashtags relacionadas a hashtag pesquisada	5
Figura 2.3: Tela principal da ferramenta Topsy	6
Figura 2.4: Tela de análise e comparação entre termos	6
Figura 2.5: Tela principal da ferramenta Keyhole	7
Figura 2.6: Resultado de busca pelo termo <i>foraDilma</i>	7
Figura 2.7: Tela principal da ferramenta Hashtracking	8
Figura 2.8: Resultado da busca pela hashtag <i>#foraDilma</i>	8
Figura 2.9: Tela principal da ferramenta Tweet Binder	9
Figura 2.10: Resultados da busca pela hashtag <i>#foraDilma</i>	9
Figura 2.11: Visualização dos termos mais frequentes por região	9
Figura 2.12: Tela principal da ferramenta tagboard	10
Figura 2.13: Resultado da busca pelo termo <i>#foraDilma</i>	10
Figura 3.1: Etapas de execução da ferramenta	13
Figura 3.2: <i>Workflow</i> de obtenção dos <i>tweets</i> , tratamento e armazenamento .	15
Figura 3.3: Etapa de obtenção dos <i>tweets</i>	16

Figura 3.4: Configuração do nó <i>Twitter API Connector</i> na etapa de Obtenção dos <i>tweets</i>	16
Figura 3.5: Configuração do nó <i>Twitter Search</i> na etapa de Obtenção dos <i>tweets</i>	17
Figura 3.6: Configuração do nó <i>Twitter Users</i> na etapa de Obtenção dos <i>tweets</i>	18
Figura 3.7: Etapa de extração e tratamento das <i>hashtags</i>	18
Figura 3.8: Configuração do nó <i>Java Snippet</i> na etapa de extração e tratamento das <i>hashtags</i>	19
Figura 3.9: Configuração do nó <i>Case Converter</i> na etapa de extração e tratamento das <i>hashtags</i>	20
Figura 3.10: Configuração do nó <i>String to Date/Time</i> na etapa de extração e tratamento das <i>hashtags</i>	21
Figura 3.11: Etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	21
Figura 3.12: Configuração do nó <i>Table Creator</i> na etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	22
Figura 3.13: Configuração do nó <i>Insert Column Header</i> na etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	22
Figura 3.14: Configuração do nó <i>Database Connector</i> na etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	23
Figura 3.15: Configuração do nó <i>Database Writer</i> na etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	23
Figura 3.16: Configuração do nó <i>Database Reader</i> na etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	24
Figura 3.17: Configuração do nó <i>Database Writer</i> na etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	24
Figura 3.18: Configuração do nó <i>Database Drop Table</i> na etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	24

Figura 3.19: Configuração do nó <i>Database Connector</i> na etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	25
Figura 3.20: Configuração do nó <i>Database Reader</i> na etapa de armazenamento dos <i>tweets</i> e <i>hashtags</i>	25
Figura 3.21: <i>Workflow</i> para geração de lista de termos próximos	27
Figura 3.22: Etapa de preparação e processamento dos dados	28
Figura 3.23: Configuração do nó <i>Strings To Document</i> no <i>workflow</i> de geração de lista de termos próximos	29
Figura 3.24: Configuração do nó <i>Punctuation Erasure</i> no <i>workflow</i> de geração de lista de termos próximos	30
Figura 3.25: Configuração do nó <i>Bag of Words</i> no <i>workflow</i> de geração de lista de termos próximos	31
Figura 3.26: Resultado da execução do nó <i>Bag of Words</i> no <i>workflow</i> de geração de lista de termos próximos	31
Figura 3.27: Configuração do nó <i>Column Filter</i> no <i>workflow</i> de geração de lista de termos próximos	31
Figura 3.28: Resultado da execução do nó <i>Column Filter</i> no <i>workflow</i> de geração de lista de termos próximos	31
Figura 3.29: Configuração do nó <i>GroupBy</i> no <i>workflow</i> de geração de lista de termos próximos	32
Figura 3.30: Configuração do nó <i>Distance Matrix Calculate</i> no <i>workflow</i> de geração de lista de termos próximos	33
Figura 3.31: Configuração do nó <i>Distance Matrix Pair Extractor</i> no <i>workflow</i> de geração de lista de termos próximos	33
Figura 3.32: <i>Workflow</i> para geração do gráfico de frequências de <i>hashtags</i>	35
Figura 3.33: Etapa de preparação e processamento dos dados	35

Figura 3.34: Configuração do nó <i>Value Counter</i> no <i>workflow</i> de geração do gráfico de frequências de hashtags	36
Figura 3.35: Resultado da execução do nó <i>Value Counter</i> no <i>workflow</i> de geração do gráfico de frequências de hashtags	36
Figura 3.36: Configuração do nó <i>RowID</i> no <i>workflow</i> de geração do gráfico de frequências de hashtags	37
Figura 3.37: Configuração do nó <i>Sorter</i> no <i>workflow</i> de geração do gráfico de frequências de hashtags	38
Figura 3.38: Etapa de exibição do resultado do <i>workflow</i> de geração do gráfico de frequências de hashtags	38
Figura 3.39: Seção de configuração de opções do nó <i>JavaScript Bar Chart</i> para geração da lista de frequência	39
Figura 3.40: Seção de configuração de opções de plotagem do nó <i>JavaScript Bar Chart</i> para geração da lista de frequência	39
Figura 3.41: <i>Workflow</i> da geração da Matriz de coocorrência	41
Figura 3.42: Etapa de preparação e processamento dos dados	42
Figura 3.43: Configuração do nó <i>Term co-occurrence counter</i> no <i>workflow</i> de geração da Matriz de coocorrência	43
Figura 3.44: Resultado da execução do nó <i>Term co-occurrence counter</i> no <i>workflow</i> de geração da Matriz de coocorrência	43
Figura 3.45: Configuração do nó <i>Term to String</i> no <i>workflow</i> de geração da Matriz de coocorrência	44
Figura 3.46: Configuração do nó <i>String to Term</i> no <i>workflow</i> de geração da Matriz de coocorrência	44
Figura 3.47: Configuração do nó <i>Column Combiner</i> no <i>workflow</i> de geração da Matriz de coocorrência	44

Figura 3.48: Resultado da execução do nó <i>String to Term</i> no <i>workflow</i> de geração da Matriz de coocorrência	45
Figura 3.49: Configuração do nó <i>Column Filter</i> no <i>workflow</i> de geração da Matriz de coocorrência	45
Figura 3.50: Configuração do nó <i>GroupBy</i> no <i>workflow</i> de geração da Matriz de coocorrência	46
Figura 3.51: Seção <i>Manual Aggregation</i> da configuração do nó <i>GroupBy</i> no <i>workflow</i> de geração da Matriz de coocorrência	46
Figura 3.52: Resultado da execução do nó <i>GroupBy</i> no <i>workflow</i> de geração da Matriz de coocorrência	47
Figura 3.53: Configuração do nó <i>Column Filter</i> no <i>workflow</i> de geração da Matriz de coocorrência	48
Figura 3.54: Resultado da execução do nó <i>Column Filter</i> no <i>workflow</i> de geração da Matriz de coocorrência	48
Figura 3.55: Configuração do nó <i>Java Snippet</i> no <i>workflow</i> de geração da Matriz de coocorrência	48
Figura 3.56: Resultado da execução do nó <i>Java Snippet</i> no <i>workflow</i> de geração da Matriz de coocorrência	49
Figura 3.57: Configuração do nó <i>Column Filter</i> no <i>workflow</i> de geração da Matriz de coocorrência	49
Figura 3.58: Resultado da execução do nó <i>Column Filter</i> no <i>workflow</i> de geração da Matriz de coocorrência	49
Figura 3.59: Configuração do nó <i>Pivoting</i> no <i>workflow</i> de geração da Matriz de coocorrência	50
Figura 3.60: Resultado da execução do nó <i>Pivoting</i> no <i>workflow</i> de geração da Matriz de coocorrência	50

Figura 3.61: Configuração do nó <i>Create Collection Column</i> no <i>workflow</i> de geração da Matriz de coocorrência	51
Figura 3.62: Resultado da execução do nó <i>Create Collection Column</i> no <i>workflow</i> de geração da Matriz de coocorrência	51
Figura 3.63: Resultado da execução do nó <i>RowID</i> no <i>workflow</i> de geração da Matriz de coocorrência	52
Figura 3.64: Etapa de exibição da matriz de coocorrência no <i>workflow</i> de geração da Matriz de coocorrência	52
Figura 3.65: Seção de opções de plotagem do nó <i>HeatMap</i> no <i>workflow</i> de geração da Matriz de coocorrência	53
Figura 3.66: Seção de configuração de eixos do nó <i>HeatMap</i> no <i>workflow</i> de geração da Matriz de coocorrência	53
Figura 3.67: Seção de opções da configuração do nó <i>Heatmap</i> no <i>workflow</i> de geração da Matriz de coocorrência	53
Figura 3.68: <i>Workflow</i> de aplicação da regra de associação	55
Figura 3.69: Etapa de preparação e processamento dos dados do <i>workflow</i> de aplicação da regra de associação	56
Figura 3.70: Configuração do nó <i>Column Filter</i> no <i>workflow</i> de aplicação da regra de associação	57
Figura 3.71: Resultado da execução do nó <i>Column Filter</i> no <i>workflow</i> de aplicação da regra de associação	57
Figura 3.72: Configuração do nó <i>RowID</i> no <i>workflow</i> de aplicação da regra de associação	58
Figura 3.73: Resultado da execução do nó <i>RowID</i> no <i>workflow</i> de aplicação da regra de associação	58

Figura 3.74: Configuração do nó <i>Column Filter</i> no <i>workflow</i> de aplicação da regra de associação	58
Figura 3.75: Resultado da execução do nó <i>Column Filter</i> no <i>workflow</i> de aplicação da regra de associação	58
Figura 3.76: Configuração do nó <i>Document Data Extractor</i> no <i>workflow</i> de aplicação da regra de associação	59
Figura 3.77: Resultado da execução do nó <i>Document Data Extractor</i> no <i>workflow</i> de aplicação da regra de associação	59
Figura 3.78: Configuração do nó <i>GroupBy</i> no <i>workflow</i> de aplicação da regra de associação	60
Figura 3.79: Resultado da execução do nó <i>GroupBy</i> no <i>workflow</i> de aplicação da regra de associação	60
Figura 3.80: Etapa de exibição do resultado no <i>workflow</i> de aplicação da regra de associação	60
Figura 3.81: Configuração do nó <i>Association Rule Leaner</i> no <i>workflow</i> de aplicação da regra de associação	61
Figura 4.1: Gráfico de frequência	69
Figura 4.2: Matriz de coocorrência	71

Lista de Tabelas

Tabela 2.1: Comparação entre ferramentas de análise de <i>hashtags</i>	11
Tabela 3.1: Exemplos do dicionário	34
Tabela 4.1: Estrutura e exemplos de dados do <i>tweet</i>	64
Tabela 4.2: Estrutura e exemplos de dados do usuário	66
Tabela 4.3: Exemplos do campo <i>hashtags</i>	67
Tabela 4.4: Resultado da aplicação de regra de associação	72

Lista de Abreviaturas e Siglas

API	Application Programming Interface
ETL	Extract Transform Load
KNIME	Konstanz Information Miner
PMDB-RJ	Partido do Movimento Democrático Brasileiro - Rio de Janeiro
PMDB-SP	Partido do Movimento Democrático Brasileiro - São Paulo
PSDB	Partido da Social Democracia Brasileira
PT	Partido dos Trabalhadores

Sumário

Agradecimentos	i
Resumo	iii
Abstract	iv
Lista de Figuras	v
Lista de Tabelas	xii
Lista de Abreviaturas e Siglas	xiii
1 Introdução	1
2 Trabalhos Relacionados	4
2.1 Sistemas Semelhantes	4
2.1.1 Hashtagify	4
2.1.2 Topsy	5
2.1.3 Keyhole	6
2.1.4 HashTracking	7
2.1.5 Tweetbinder	8

2.1.6	TrendsMap	9
2.1.7	TagBoard	10
2.2	Análise Comparativa	10
3	Proposta do Workflow	12
3.1	Proposta Geral	12
3.2	Ferramenta KNIME	12
3.3	Especificação do <i>Workflow</i>	13
3.3.1	Obtenção dos <i>tweets</i> , tratamento e armazenamento	14
3.3.1.1	Obtenção dos <i>tweets</i>	15
3.3.1.2	Extração e tratamento das <i>hashtags</i>	18
3.3.1.3	Armazenamento dos <i>tweets</i> e <i>hashtags</i>	21
3.3.2	Obtenção dos dados armazenados	25
3.3.3	Geração do dicionário de termos	26
3.3.3.1	Preparação e processamento dos dados	27
3.3.4	Geração do gráfico de frequências de <i>hashtags</i>	34
3.3.4.1	Preparação e processamento dos dados	35
3.3.4.2	Exibição do Resultado	37
3.3.5	Geração da lista de frequência	39
3.3.6	Matriz de coocorrência	40
3.3.6.1	Preparação e processamento dos dados	41
3.3.6.2	Exibição do resultado	52
3.3.7	Aplicação da regra de associação	54

3.3.7.1	Preparação e processamento dos dados	55
3.3.7.2	Exibição do Resultado	60
4	Resultados e Discussão	62
4.1	Descrição geral dos experimentos	62
4.2	Dados do Twitter	63
4.3	Resultados	68
4.4	Discussão	72
5	Conclusão e Trabalhos Futuros	75
Referências		77

Capítulo 1

Introdução

Com a evolução tecnológica, dispositivos eletrônicos, como notebooks, tablets, smartphones e etc., se tornam cada vez mais presentes no cotidiano. Além dos benefícios diretos do seu uso, estes dispositivos geram grandes volumes de dados com potencial ainda pouco explorado.

Dentre as diversas fontes de informação, ganham destaque as redes sociais, que passaram a fazer parte da vida da sociedade, e é através destas que os indivíduos vêm manifestando seus interesses, opiniões e críticas. Dentre as maiores redes sociais destaca-se o Twitter [53], um *microblogging* que permite aos usuários criarem e compartilharem ideias e informações de forma instantânea. Essas informações são veiculadas através de *tweets*, uma mensagem criada e compartilhada pela rede. O *tweet* possui diversas formas de comunicação, fazendo uso de certas funcionalidades para indexação e categorização da informação na rede social, uma delas é a *hashtag*. Esta consiste de uma cerquilha ("#") acompanhada de um ou mais termos concatenados. Assim, a *hashtag* permite o fácil acompanhamento de conteúdo, evitando com que usuários se percam em meio a imensidão de *tweets* na rede. Consequentemente, uma *hashtag* acaba por carregar uma enorme carga semântica a respeito de seus temas ou assuntos. *Hashtags* podem ser utilizadas em ações de marketing por empresas, criando um canal onde a marca pode interagir com o público, por exemplo a partir de concursos ou campanhas promocionais, buscando a atenção de consumidores e promovendo a marca. Portanto, várias organizações monitoram

hashtags que, de alguma forma, estão relacionadas a sua marca, ou ainda procuram por padrões entre estas e outras *hashtags* não associadas diretamente à marca.

Em consequência da grande quantidade de dados e procura das empresas, faz-se necessário o desenvolvimento de novas ferramentas que empregam técnicas de extração de conhecimento a partir desses dados armazenados. Tais ferramentas se tornaram indispensáveis, face à inviabilidade de um processo de análise manual. Dentre essas técnicas está a mineração de dados, onde se tem como objetivo a identificação de padrões relevantes em grandes volumes de dados e se permite a busca, de forma prática, das informações.

Neste trabalho, desenvolveu-se uma ferramenta de mineração de dados que possibilita estudos e análise da difusão de *hashtags* no Twitter, no qual se utiliza a ferramenta KNIME para a construção e obtenção de resultados. O estudo de caso, o qual aplicou-se a ferramenta, teve como base os acontecimentos políticos ocorridos entre novembro e dezembro do ano de 2015. Neste período, o Brasil esteve em meio a uma crise econômica e uma disputa de poder entre políticos da base aliada do governo Dilma Rousseff e a oposição chamou a atenção da população em geral. Diversas atitudes por parte da presidente, de seu partido, o PT, do então presidente da Câmara dos Deputados, Eduardo Cunha (PMDB-RJ), do vice-presidente Michel Temer (PMDB-SP) e de alguns políticos do PSDB, principal partido de oposição, protagonizaram diversos escândalos e trocas de acusações. Neste contexto, foram escolhidas as *hashtags* *#foraDilma* e *#foraCunha* para serem estudadas e acompanhadas por meio de mineração de dados com objetivo de analisar sua difusão e comportamento, além de visualizar possíveis manifestações de apoio ou repúdio aos políticos e suas atitudes.

Nas seções seguintes se encontram o desenvolvimento do trabalho. Na segunda seção apresenta-se um estudo sobre as ferramentas encontradas no mercado que realizam análises utilizando *tweet* e *hashtags*. Na terceira seção pode ser vista a proposta de desenvolvimento da análise é descrita, onde são compreendidas a ferramenta utilizada, as técnicas de tratamento e análise escolhidas e as soluções aplicadas. Na quarta seção encontram-se os resultados e discussões que puderam ser obtidas atra-

vés da análise. Na quinta seção é exibida uma conclusão do trabalho e possíveis trabalhos futuros com o tema.

Capítulo 2

Trabalhos Relacionados

Este capítulo oferece uma visão geral das ferramentas de análise de *hashtags* oriundas dos dados do Twitter. O objetivo é de apontar as funcionalidades de cada ferramenta e possibilitar a comparação entre elas. O capítulo é dividido em seções, onde cada seção apresenta uma ferramenta e descreve suas principais características.

2.1 Sistemas Semelhantes

2.1.1 Hashtagify

Hashtagify [15] é uma das ferramentas mais completas do mercado com foco total na análise das *hashtags*, permitindo a utilização de funcionalidades como correlação entre *hashtags*, melhores horários e idiomas para utilização de uma determinada *hashtag* e sua popularidade, *hashtags* relacionadas ao termo pesquisado e visualização através de grafo ou tabela. A aquisição de todas essas funcionalidades pode se tornar inviável pelos altos valores dos planos de utilização [17]. Embora exista um plano gratuito, esse não possui muitas funcionalidades. Além da utilização online, essa ferramenta possui uma API [16] que possibilita a incorporação das funcionalidades da ferramenta em sistemas próprios com acesso a mais de 50 milhões de *hashtags* em sua enciclopédia [18]. Nas figuras 2.1 e 2.2 podem ser vistos exemplos

2.1. SISTEMAS SEMELHANTES

5

da utilização da ferramenta.

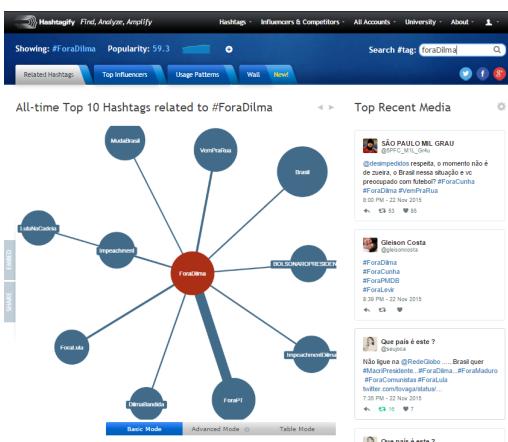


Figura 2.1: Tela principal da ferramenta Hashtagfy

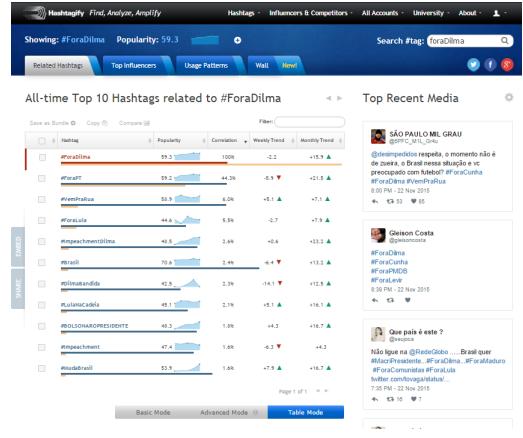


Figura 2.2: Dez maiores hashtags relacionadas a hashtag pesquisada

2.1.2 Topsy

Topsy [46] é um parceiro certificado pelo Twitter e mantém uma grande quantidade de *tweets*, estimada em centenas de milhões, armazenada desde 2006, início das atividades do Twitter.

É considerada uma enciclopédia, por possuir toda a história do Twitter e disponibilizá-la para busca de *tweets* de forma rápida, simples e gratuita. Essa busca é facilitada por possuir diversos filtros, como data e idioma, que possibilitam a visualização dos *tweets* e recursos relacionados, tais como fotos e vídeos. Também possui uma comparação entre *hashtags*, onde pode ser utilizada uma busca avançada que possibilita o uso de operadores especiais, como operadores lógicos e referências a usuários.

Além da ferramenta em si, a Topsy possui uma API [47] que possibilita a utilização de todo seu histórico de *tweets* em uma aplicação própria. Essa API tem um custo de utilização, que é calculado pela quantidade de *tweets* utilizados.

Nas figuras 2.3 e 2.4 podem ser vistos exemplos da utilização da ferramenta.

2.1. SISTEMAS SEMELHANTES

6



Figura 2.3: Tela principal da ferramenta Topsy

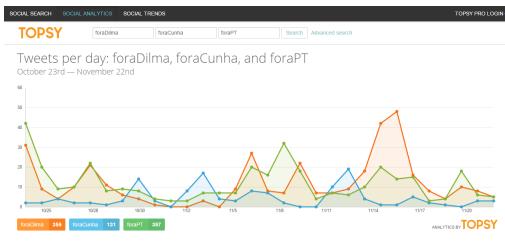


Figura 2.4: Tela de análise e comparação entre termos

2.1.3 Keyhole

A Keyhole [24] é uma ferramenta que possui foco em rastreamento de *hashtags*. Com ela é possível visualizar o número de postagens com a *hashtag* pesquisada, número de usuários que utilizaram a *hashtag* e o alcance gerado. Apresenta *dashboards* com a evolução do número de postagens de acordo com a hora, porcentagem entre postagens originais, *retweets* e *replies*, lista da postagem mais *retweetada*, dos usuários mais influentes e recentes, além de uma nuvem de termos com *hashtags* relacionadas com a *hashtag* pesquisada.

Também possui planos pagos [25] com funcionalidades mais completas, inclusive acesso a dados históricos.

Além da ferramenta, possui uma API [25] para integração com sistemas internos. Essa API está disponível nos planos: Enterprise, Agency e Partner.

Nas figuras 2.5 e 2.6 podem ser vistos exemplos da utilização da ferramenta.

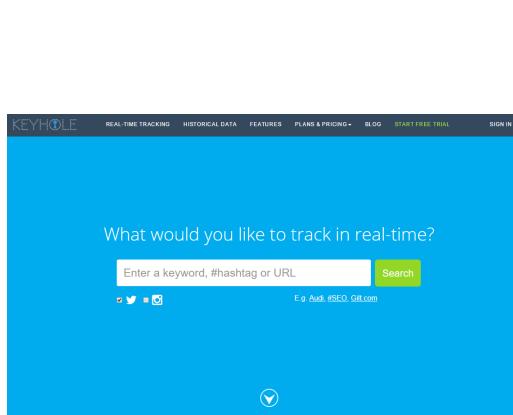


Figura 2.5: Tela principal da ferramenta Keyhole

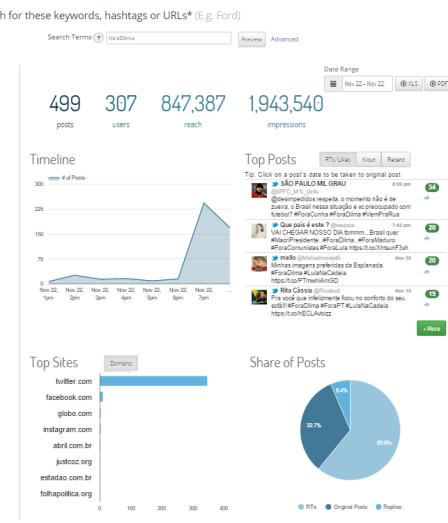


Figura 2.6: Resultado de busca pelo termo *foraDilma*

2.1.4 HashTracking

O Hashtracking [19] é uma ferramenta de análise de *hashtags*, que podem ser oriundas do Twitter ou do Instagram e possui bastante semelhança com a ferramenta Keyhole, com a principal diferença de que o usuário precisa fazer *login* com sua conta do Twitter através da ferramenta para que ela funcione.

Mostra porcentagem entre os *tweets* originais, *replies* e *retweets*, o alcance, os usuários mais influentes, com base no número de seguidores, e os que mais utilizaram a *hashtag*.

Possui planos [20] para rastreamento de *hashtags*, os quais são diferenciados basicamente pela quantidade de *hashtags* rastreadas.

Nas figuras 2.7 e 2.8 podem ser vistos exemplos da utilização da ferramenta.



Figura 2.7: Tela principal da ferramenta Hashtracking

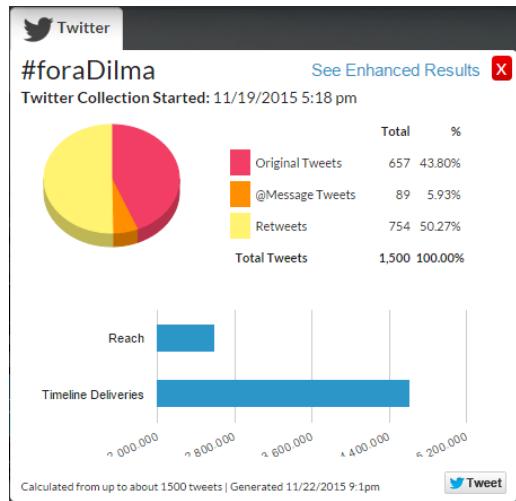


Figura 2.8: Resultado da busca pela hashtag #fora-Dilma

2.1.5 Tweetbinder

O TweetBinder [51] é uma ferramenta que também necessita que o usuário faça *login* com sua conta do Twitter e permite a análise de uma ou mais *hashtags*, menções ou palavras-chave. É bastante semelhante às ferramentas já apresentadas.

Ele mostra percentual entre *retweets*, *replies* e mensagens originais, usuários mais influentes, e também as imagens que fazem parte dos *posts* dos usuários.

Também possui planos [52] para rastreamento de *hashtags*, os quais são diferenciados basicamente pela quantidade de *hashtags* rastreadas, semelhante aos já mostrados.

Nas figuras 2.9 e 2.10 podem ser vistos exemplos da utilização da ferramenta.

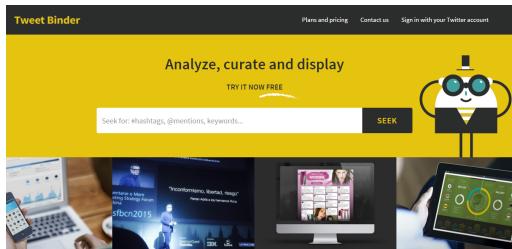


Figura 2.9: Tela principal da ferramenta Tweet Binder



Figura 2.10: Resultados da busca pela hashtag #fora-Dilma

2.1.6 TrendsMap

O TrensMap [49] é uma ferramenta um pouco diferente das já citadas nesta seção, esta mostra quais são os termos, sejam eles *hashtags*, menções, ou palavras-chave, mais frequentes por região. Possui uma visualização em forma de mapa, onde o termo fica localizado sobre a região. Ao fazer *login* no Twitter pela ferramenta é possível clicar em qualquer termo localizado no mapa que será aberto uma janela com os últimos *posts* daquele termo na região.

Também possui planos [50] com diversas funcionalidades extras, como outros tipos de visualizações e uso de API própria. A API possibilita a análise dos dados desde o início do Twitter, em 2006 e recebimento de notificações a cada 5 minutos.

Na figura 2.11 pode ser visto exemplo da utilização da ferramenta.

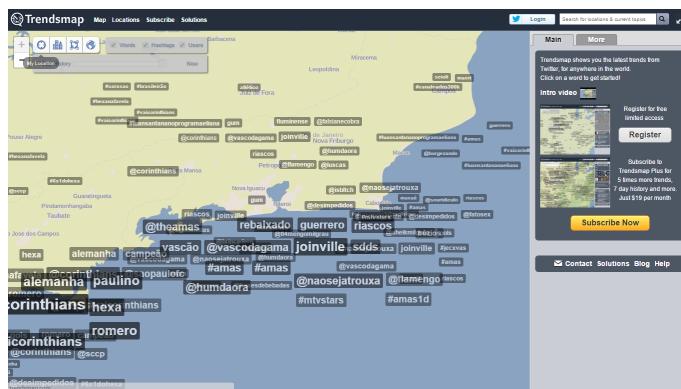


Figura 2.11: Visualização dos termos mais frequentes por região

2.1.7 TagBoard

O Tagboard [42] é uma ferramenta de análise de *hashtags* não só do Twitter, mas também do Facebook, Instagram, Google+, Vine e Flickr. Permite filtragem por plataforma e mostra diretamente os *posts* onde foram encontrados a *hashtag* pesquisada.

Possui análise de sentimento dos *posts*, separados em positivos, neutros e negativos.

Também possui planos [43] onde é possível criar soluções personalizadas de acordo com a necessidade do cliente.

Nas figuras 2.12 e 2.13 podem ser vistos exemplos da utilização da ferramenta.



Figura 2.12: Tela principal da ferramenta tagboard

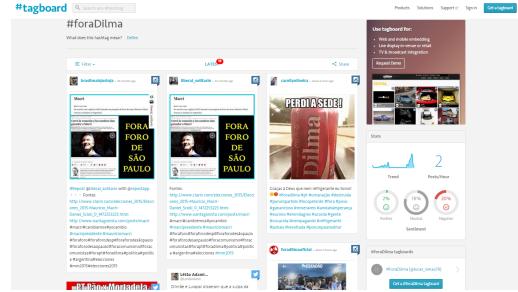


Figura 2.13: Resultado da busca pelo termo #foraDilma

2.2 Análise Comparativa

Essa seção tem como objetivo apresentar uma análise comparativa entre as ferramentas citadas na seção anterior com base em suas funcionalidades, apresentada na Tabela 2.1. O 'X' representa que há a funcionalidade na ferramenta. A análise apresentada foi baseada na versão gratuita de cada ferramenta.

Tabela 2.1: Comparaçāo entre ferramentas de análise de *hashtags*

	Hashtagify	Topsy	Keyhole	Hashtracking	TweetBinder	TrendsMap	TagBoard
Número de <i>posts</i> por <i>hashtag</i>	-	x	x	x	x	-	-
Número usuários por <i>hashtag</i>	-	-	x	-	x	-	-
Frequência de <i>posts</i> por dia/hora	-	x	x	-	x	-	x
Análise de sentimento dos <i>posts</i>	-	x	-	-	-	-	x
<i>Hashtags</i> relacionadas	x	-	x	-	-	-	-
Visualização Georreferenciada	-	-	x	-	-	x	-
Perfis mais influentes	-	x	x	x	x	-	-
Correlação entre <i>hashtags</i>	x	-	-	-	-	-	-
Número de <i>retweets</i> e replicações	-	-	x	x	x	-	-

Comparando as ferramentas, é visto que aquela com maior número de funcionalidades é a Keyhole, porém a mesma não dispõe de funcionalidades de maior complexidade, como a análise de sentimentos dos *posts* e correlação entre *hashtags*. Para tais, Topsy e TagBoard, se destacam ao realizarem uma análise de sentimento dos *posts*, e Hashtagify, ao realizar a correlação entre *hashtags*.

Conclui-se que nenhuma das ferramentas analisadas possuem funcionalidades de mineração de dados para busca de padrões entre *hashtags*, motivando a criação de tal objeto de pesquisa.

Capítulo 3

Proposta do Workflow

3.1 Proposta Geral

Esse trabalho tem como objetivo propor uma ferramenta de mineração de dados, onde dados oriundos do Twitter podem ser tratados e analisados através de suas *hashtags*. Para tal, *workflows* [35], sequência de passos para automatização e execução de um processo, onde a informação transita entre os nós participantes criando um "fluxo de trabalho", são implementados através da ferramenta KNIME, apresentada na próxima seção.

Foram criados *workflows* [35] para obtenção dos dados, extração das *hashtags*, armazenamento, tratamento e seleção dos dados, criação de gráfico de frequências, criação de matriz de coocorrência e utilização da regra de associação [62, 60] entre tais *hashtags*.

3.2 Ferramenta KNIME

KNIME [27] é uma ferramenta *open source*, desenvolvida a partir de 2004 pela Universidade de Konstanz, situada na Alemanha, que possibilita análise de dados, elaboração de relatórios e integração de plataformas [26]. É uma ferramenta de implantação rápida, fácil de escalar e intuitiva de se usar.

Pode integrar vários tipos de componentes para análise de dados, como componentes de aprendizado de máquina e mineração de dados através do seu conceito de *pipeline* de dados modular. Neste, o processamento é subdividido em etapas, ou módulos, possibilitando execução simultânea de módulos, que geralmente, possuem uma entrada e saída, e a possibilidade de configurações personalizadas.

O KNIME permite o processamento de grandes volumes de dados através de métodos de mineração de texto, de imagem, algoritmos de aprendizado de máquina, utilização de linguagem R, Java e pré-processamento de dados (ETL). Tem sido usado principalmente em pesquisa farmacêutica [59], quimioterapia e Bioinformática, mas também é usado em outras áreas como análise de dados de alto desempenho e de outras áreas da indústria, inteligência de negócios e análise de dados financeiros [28].

3.3 Especificação do *Workflow*

Esta seção apresenta a ferramenta desenvolvida para obtenção dos resultados apresentados na quarta seção. Tal ferramenta compreende-se pelo conjunto de etapas mencionados na figura 3.1. As etapas, com exceção da obtenção dos *tweets*, necessitam da conclusão da etapa anterior para serem executadas. Cada etapa está compreendida na execução de um ou mais *workflows*.



Figura 3.1: Etapas de execução da ferramenta

A etapa de obtenção dos *tweets* utiliza-se do *workflow* apresentado na seção 3.3.1, tem como objetivo, a partir de nós relacionados a API do *Twitter*, obter os *tweets* e dados relacionados a este, como usuário e data de criação.

A extração de *hashtag* é realizada após a obtenção dos *tweets* e está relacionada também a execução do *workflow* da seção 3.3.1.

O tratamento básico e armazenamento, também realizado através da execução *workflow* da seção 3.3.1, padroniza as *hashtags* tirando pontuações e as colocando em letras maiúsculas e as armazena junto a seus dados relacionados utilizando-se de um banco PostgreSQL.

A etapa de tratamento avançado, apresentado na seção 3.3.3, auxilia a detecção de erros de digitação em *hashtags* utilizando-se da comparação entre termos. Esta comparação retorna uma lista de pares de termos muito próximos léxica e semanticamente que podem ser utilizados para criação de um dicionário para tradução.

A seleção dos dados, apresentado na seção 3.3.5, utiliza-se da frequência de ocorrência de um termo nos dados para *rankeamento* e posterior seleção dos termos para utilização na geração dos resultados.

O gráfico de frequência, apresentado na seção 3.3.4, utiliza-se das *hashtags* selecionadas e *rankeadas* para exibição em um gráfico de barras dos termos mais frequentes encontrados.

A matriz de coocorrência, apresentado na seção 3.3.6, utiliza-se das *hashtags* selecionadas e *rankeadas* para exibição de uma matriz de calor para a exibição da frequência de coocorrência entre um par de termos.

A regra de associação, apresentado na seção 3.3.7, utiliza-se das *hashtags* selecionadas e *rankeadas* para o processamento da regra de associação exibindo os termos mais associados entre os dados.

3.3.1 Obtenção dos *tweets*, tratamento e armazenamento

Na figura 3.2 é apresentado o *workflow* completo, onde é realizada a obtenção dos *tweets*, extração das *hashtags* e armazenamento em um banco de dados.

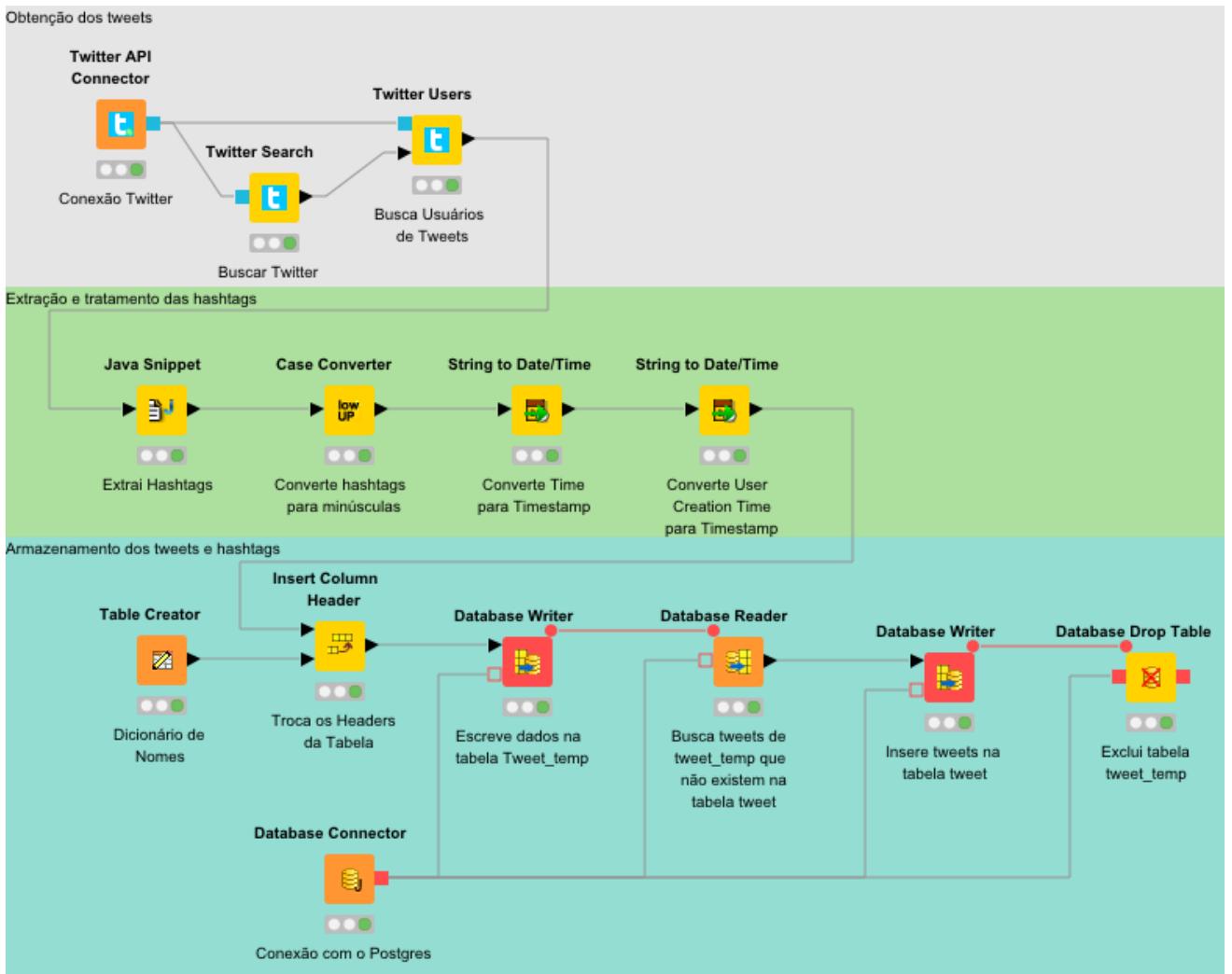


Figura 3.2: *Workflow* de obtenção dos *tweets*, tratamento e armazenamento

3.3.1.1 Obtenção dos tweets

Na figura 3.3 é apresentada a primeira etapa do *workflow*, onde é realizada a obtenção dos *tweets* que serão tratados e armazenados. São utilizados três nós da seção *Social Media/Twitter API* [29] da ferramenta KNIME [27].

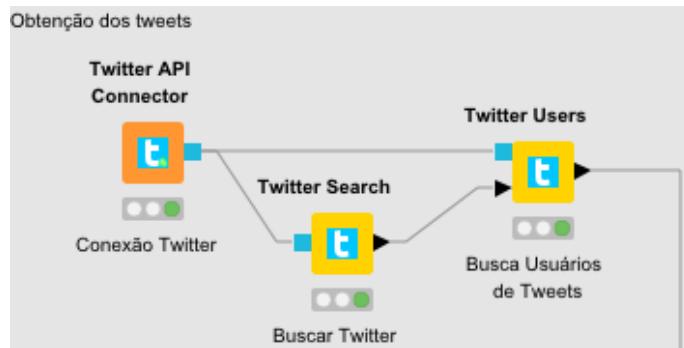


Figura 3.3: Etapa de obtenção dos *tweets*

O nó *Twitter API Connector* [54], apresentado na figura 3.4, realiza a conexão com a API do Twitter. O uso desse nó é necessário para a utilização de qualquer nó da seção *Twitter API* do KNIME [27].

Para a conexão com a API é necessário o cadastro no site *Twitter Developers* [55], onde são obtidas as chaves necessárias para a conexão com a API. Após obtidas, devem ser introduzidas nos campos abaixo, nas propriedades do nó no KNIME [27].

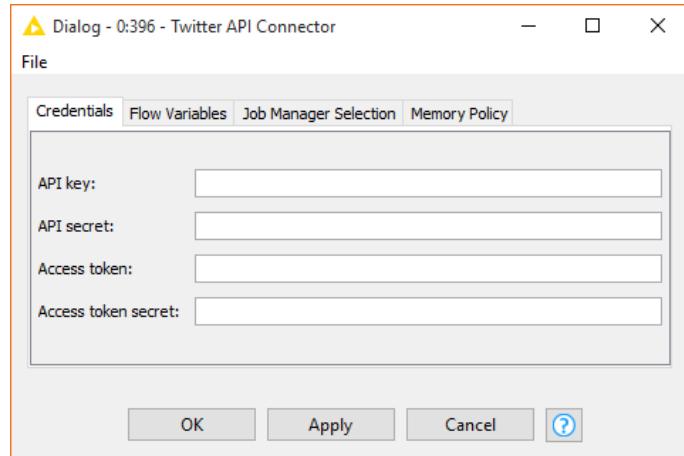


Figura 3.4: Configuração do nó *Twitter API Connector* na etapa de Obtenção dos *tweets*

O nó *Twitter Search* [57], apresentado na figura 3.5, busca no Twitter por um termo específico ou por uma expressão, onde pode ser realizada uma busca avançada por *tweets*. A API impõe uma limitação da quantidade de *tweets* obtidos, pode-se obter até 18.000 *tweets* a cada 15 minutos.

No campo *Query*, insere-se o parâmetro de busca no Twitter. Pode ser utilizado um termo, como “#foraDilma”, ou uma expressão, como “#foraDilma OR #foraCunha”.

No campo *Search for*, seleciona-se opção de busca, possibilitando a busca por *tweets* populares, mais recentes ou uma mistura de ambos.

No campo *Number of rows*, deve ser inserido o número de *tweets* esperados no retorno. A *Query* “#foraDilma OR #foraCunha” foi utilizada em conjunto com um busca mista entre *tweets* populares e recentes, onde 18.000 *tweets* era a meta de recebimento.

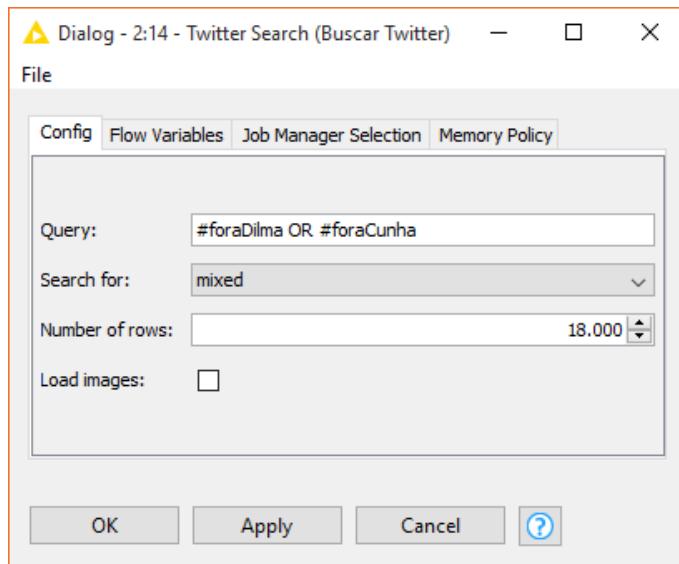


Figura 3.5: Configuração do nó *Twitter Search* na etapa de Obtenção dos *tweets*

O nó *Twitter Users* [56], apresentado na figura 3.6, possibilita a obtenção dos dados do usuário relacionado com o *tweet*.

Para execução do nó são necessários, a conexão com Twitter e o resultado da busca por *tweets* realizados nos nós anteriores. No campo *User column* é selecionado o campo onde se obtém o usuário a ser buscado no Twitter.

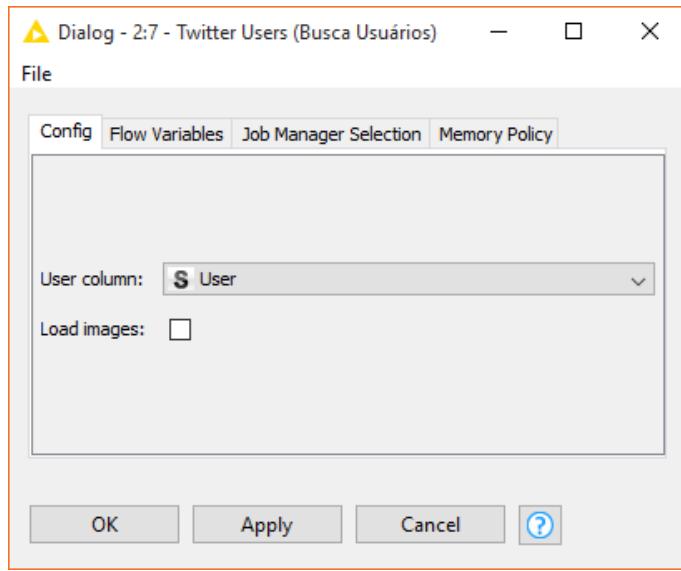


Figura 3.6: Configuração do nó *Twitter Users* na etapa de Obtenção dos *tweets*

3.3.1.2 Extração e tratamento das hashtags

A segunda etapa do *workflow*, apresentado na figura 3.7, tem como objetivo a extração das *hashtags*, tratamentos como retirada de pontuações e conversão de letras em minúsculas e conversão de tipos para inserção no banco de dados.

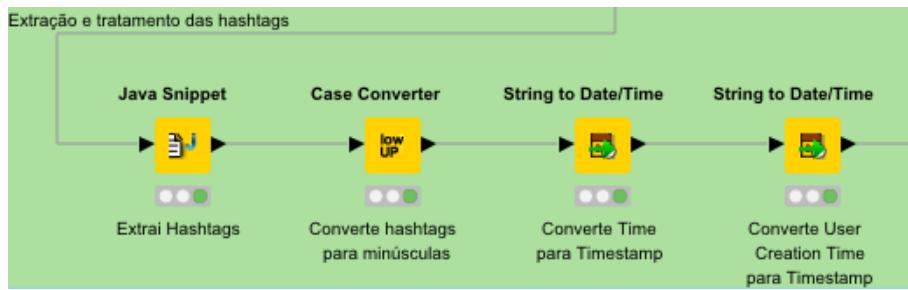


Figura 3.7: Etapa de extração e tratamento das *hashtags*

O nó que é utilizado para extração das *hashtags* e retirada de pontuações é o *Java Snippet* [23], apresentado na figura 3.8, que permite a inclusão e execução de código fonte Java no *workflow*, possibilitando a criação de novas colunas ou atualização de colunas existentes.

No código utilizado, extrai-se as *hashtags* e utiliza-se de uma função implemen-

tada para exclusão de pontuações e acentuações.

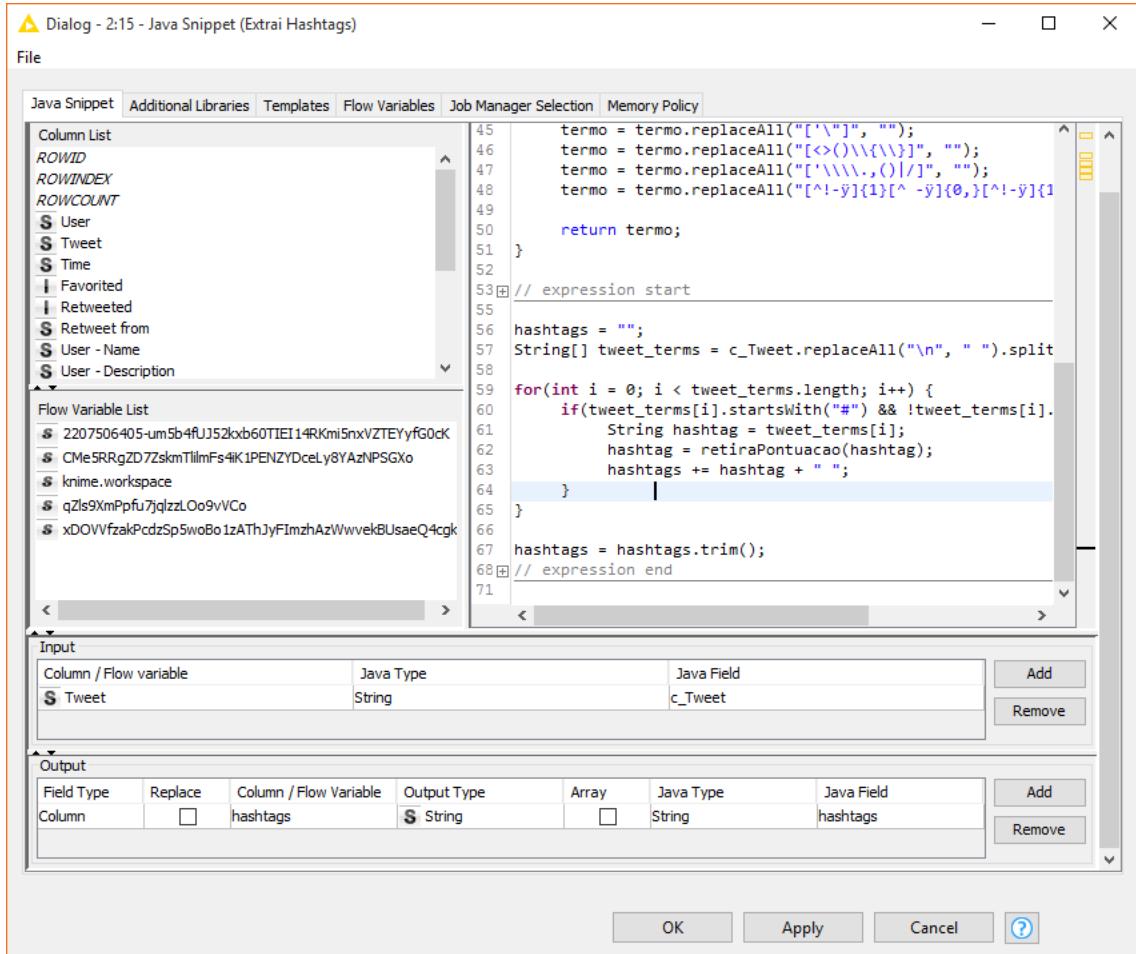


Figura 3.8: Configuração do nó *Java Snippet* na etapa de extração e tratamento das *hashtags*

O nó *Case Converter* [4], apresentado na figura 3.9, possibilita a conversão das letras de uma palavra para minúsculas ou maiúsculas. No caso, foi utilizada a conversão para minúsculas.

Para essa conversão, escolhe-se os campos a serem convertidos e o tipo de conversão como na figura.

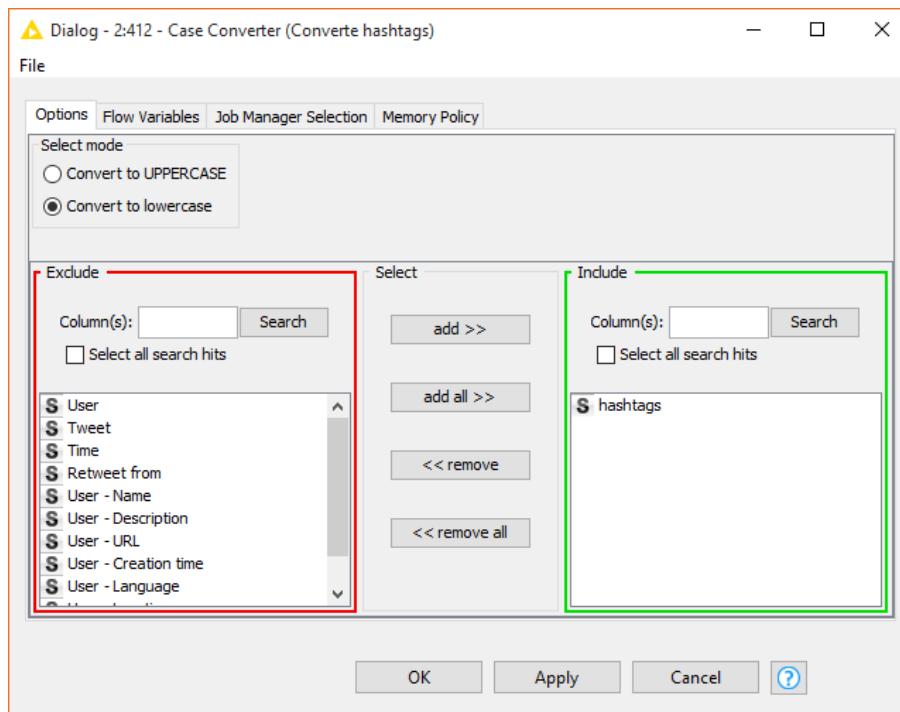


Figura 3.9: Configuração do nó *Case Converter* na etapa de extração e tratamento das *hashtags*

Para inclusão na tabela do banco de dados, é necessária a conversão dos campos *String* de data/hora oriundos do Twitter para o tipo *timestamp*, referente ao campo data/hora do Postgres.

O nó utilizado para essa conversão é o *String to Data/Time* [38], apresentado na figura 3.10,. Nele seleciona-se o campo a ser convertido e o formato de data requerido.

O campos de data do *tweet* e da criação do perfil do usuário são convertidos.

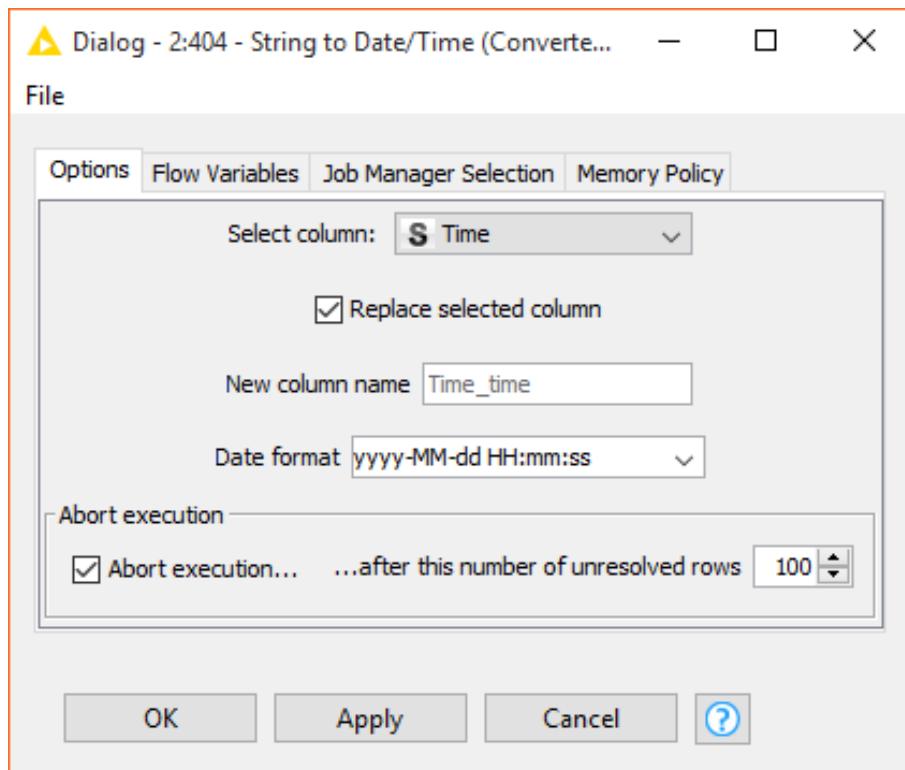


Figura 3.10: Configuração do nó *String to Date/Time* na etapa de extração e tratamento das *hashtags*

3.3.1.3 Armazenamento dos tweets e hashtags

A terceira etapa do *workflow*, apresentado na figura 3.11, possibilita a inclusão dos dados no banco de dados. Essa inclusão é realizada utilizando uma tabela temporária chamada *tweet_temp*, por ser utilizada para comparações com os dados previamente incluídos na tabela *tweet* em outras execuções do *workflow*. Após essas comparações, os dados não diferentes são incluídos na tabela *tweet*.

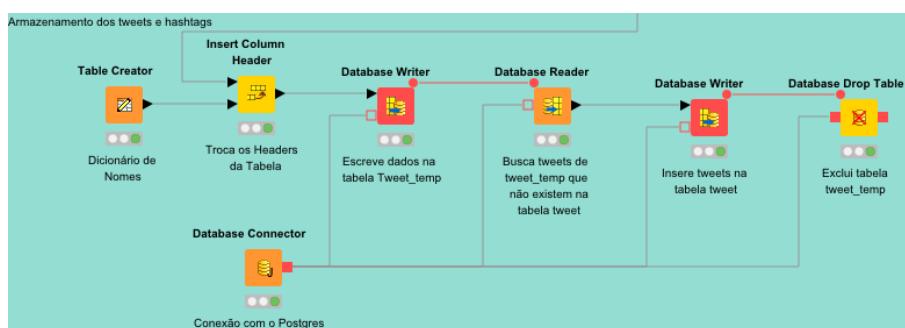


Figura 3.11: Etapa de armazenamento dos *tweets* e *hashtags*

O primeiro nó que pode ser visto é o *Table Creator* [41], apresentado na figura 3.12. Este permite a criação de um dicionário de termos que será utilizado na execução do nó *Insert Column Header* [22]. Este dicionário possui o valor do termo recebido do Twitter e sua tradução. O *Insert Column Header* [22] utiliza-se do dicionário para nomear o campo da tabela criada. Nessa etapa é criada a tabela temporária *tweet_temp*.

No *Insert Column Header* [22], apresentado na figura 3.13, para configurar a criação da tabela, no campo *Lookup Column* seleciona-se a coluna do dicionário relacionada ao termo e no campo *Value Column* para a tradução.

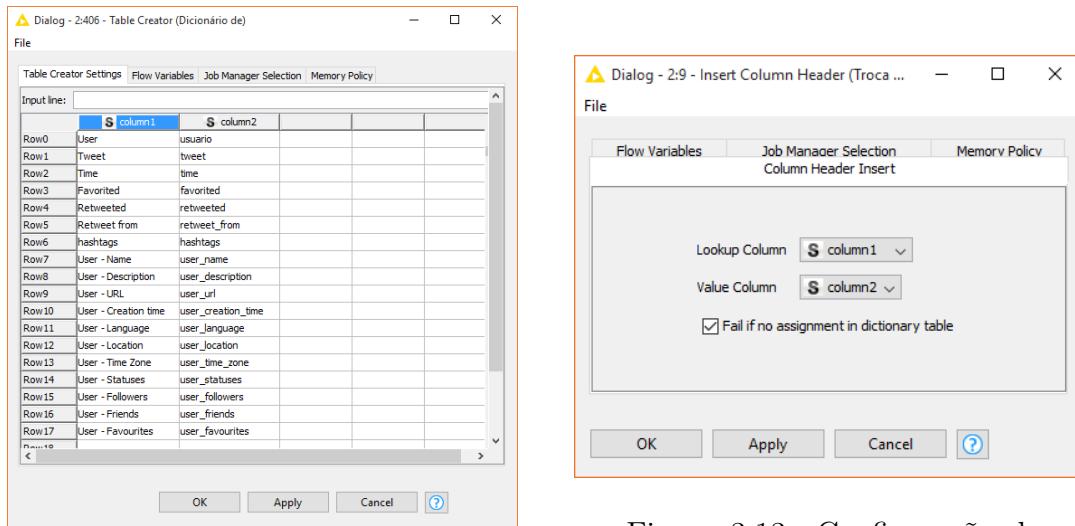


Figura 3.12: Configuração do nó *Table Creator* na etapa de armazenamento dos *tweets* e *hashtags*

Figura 3.13: Configuração do nó *Insert Column Header* na etapa de armazenamento dos *tweets* e *hashtags*

Para inclusão dos dados oriundos do Twitter na tabela temporária, o nó *Database Writer* [10], apresentado na figura 3.14, é utilizado. O nó *Database Connector* [8], apresentado na figura 3.15, é requisito necessário para execução do *Database Writer* [10].

No nó *Database Connector* [8] é criada a conexão com um banco Postgres. Para conexão ao Postgres, preenche-se o campo *Database URL* com a *String* de conexão, onde inclui-se o endereço do servidor Postgres, a porta para conexão e o nome do

banco de dados. Os campos *username* e *password* também são preenchidos, esses com usuário e senha do banco de dados, respectivamente.

Para criação da tabela no nó *Database Writer* [10] é incluído o nome da tabela a ser criada. No caso, tabela temporária *tweet_temp*.

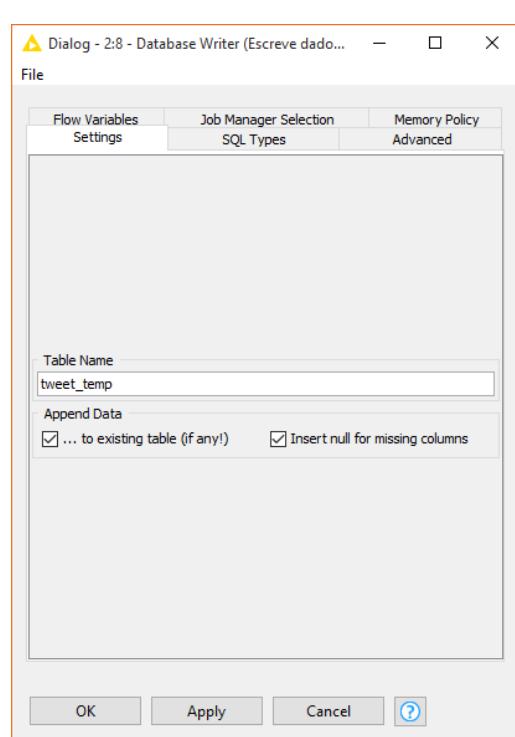
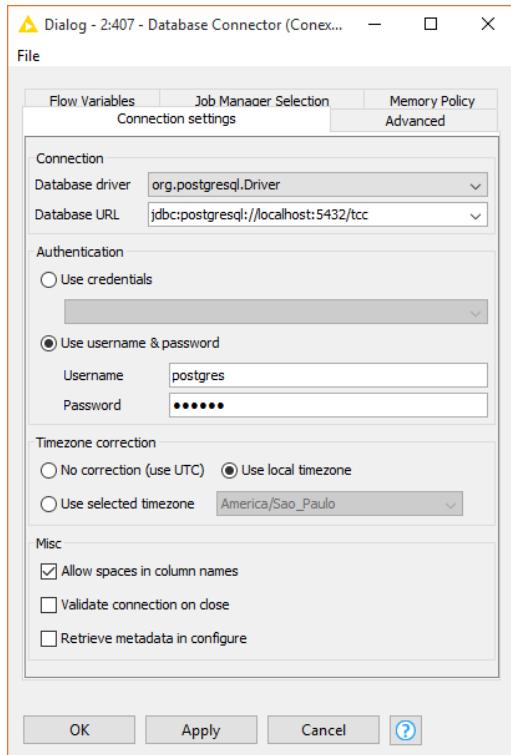


Figura 3.14: Configuração do nó *Database Connector* na etapa de armazenamento dos *tweets* e *hashtags*

Figura 3.15: Configuração do nó *Database Writer* na etapa de armazenamento dos *tweets* e *hashtags*

Para comparação entre os resultados obtidos e os previamente incluídos em execuções anteriores do *workflow* é utilizado o nó *Database Reader* [9], apresentado na figura 3.16, que tem como objetivo executar consultas em um banco de dados. No *workflow* foi utilizada uma consulta que visa retornar a diferença entre o obtido do Twitter e o armazenado na tabela *tweet*, tabela principal onde novos dados são armazenados a cada execução. Essa comparação é realizada através dos campos *time*, *usuario* e *tweet*.

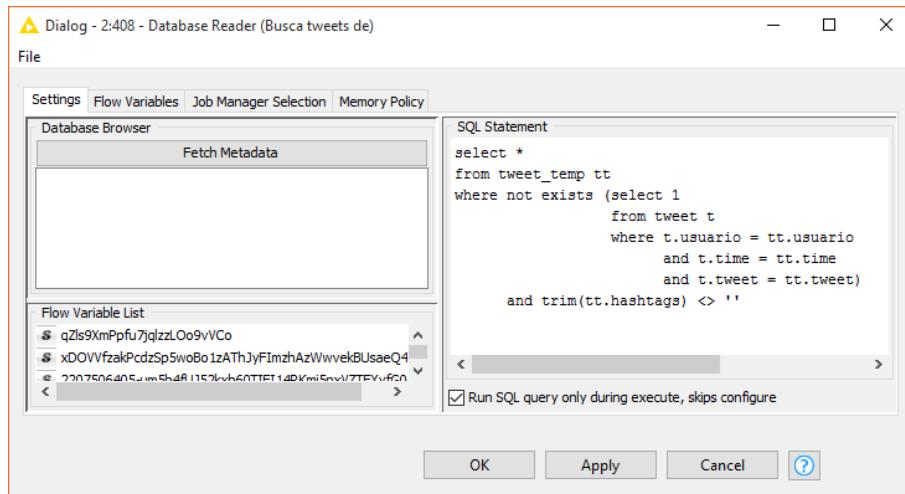


Figura 3.16: Configuração do nó *Database Reader* na etapa de armazenamento dos *tweets* e *hashtags*

A inclusão dos dados na tabela *tweet* é realizada com os dados resultantes da execução do nó anterior, utilizando o nó *Database Writer* [10], apresentado na figura 3.17.

Para finalizar o *workflow* a tabela temporária é excluída através da execução do nó *Database Drop Table* [11], apresentado na figura 3.18. Nesse nó escolhe-se a tabela a ser excluída.

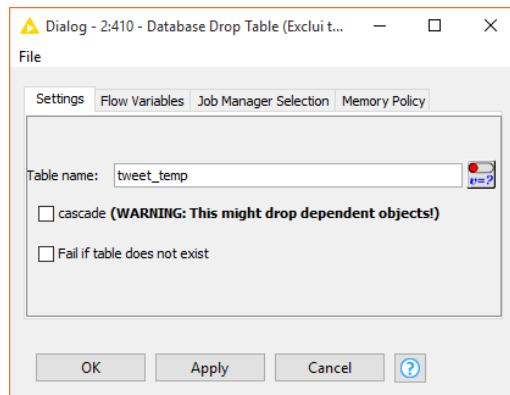
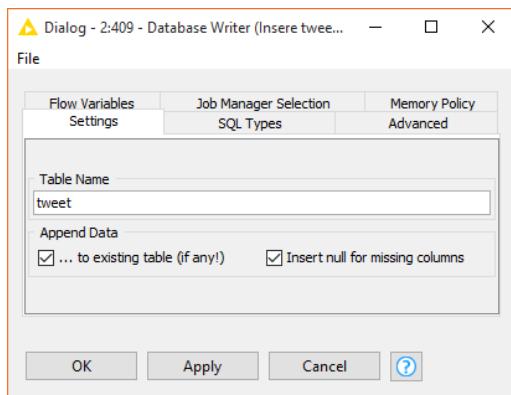


Figura 3.17: Configuração do nó *Database Writer* na etapa de armazenamento dos *tweets* e *hashtags*

Figura 3.18: Configuração do nó *Database Drop Table* na etapa de armazenamento dos *tweets* e *hashtags*

3.3.2 Obtenção dos dados armazenados

Para obtenção dos dados armazenados é utilizado o nó *Database Reader* [9], apresentado na figura 3.19, que tem como objetivo executar consultas em um banco de dados. No *workflow* foi utilizada uma consulta que visa retornar somente o campo *hashtags* da tabela *tweet*.

Para a utilização desse nó é necessária a execução prévia do nó *Database Connector* [8], apresentado na figura 3.20. Esse cria a conexão com o Postgres. Para isso, preenche-se o campo *Database URL* com a *String* de conexão, onde inclui-se o endereço do servidor Postgres, a porta para conexão e o nome do banco de dados. Os campos *username* e *password* também são preenchidos com usuário e senha do banco de dados, respectivamente. Essa etapa repete-se nos *workflows* das seções 3.3.3 a 3.3.7.

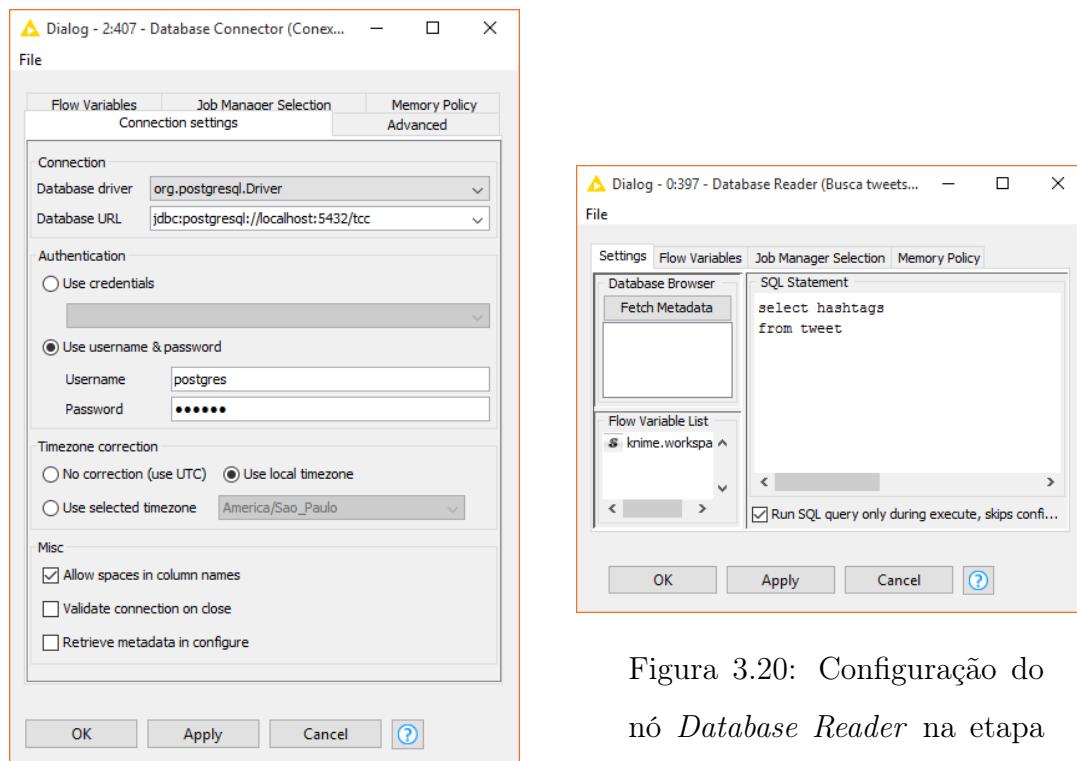


Figura 3.19: Configuração do nó *Database Connector* na etapa de armazenamento dos *tweets* e *hashtags*

Figura 3.20: Configuração do nó *Database Reader* na etapa de armazenamento dos *tweets* e *hashtags*

3.3.3 Geração do dicionário de termos

O *workflow* desta subseção objetiva a criação de um dicionário de termos. O dicionário é uma tabela na estrutura “De/Para” onde um termo deve ser traduzido para outro termo muito próximo léxica e semanticamente. Para tal criação, usa-se a distância de Levenshtein [1, 61] para o cálculo desta proximidade e análise posterior para acerto nas traduções.

A distância de Levenshtein [1, 61] entre dois termos é obtida através do número mínimo de operações utilizadas para transformar um termo no outro, sendo estas operações, a adição, remoção ou substituição de caracteres.

Por exemplo, entre as palavras *kitten* (gato) e *sitting* (sentando-se) a distância de Levenshtein [1, 61] é 3, pois com três operações consegue-se transformar um termo no outro:

I kitten

II sitten (substituição de 'k' por 's')

III sittin (substituição de 'e' por 'i')

IV sitting (inserção de 'g' no final)

O *workflow*, apresentado na figura 3.21, exibe as fases de obtenção dos dados, de preparação e processamento desses dados.

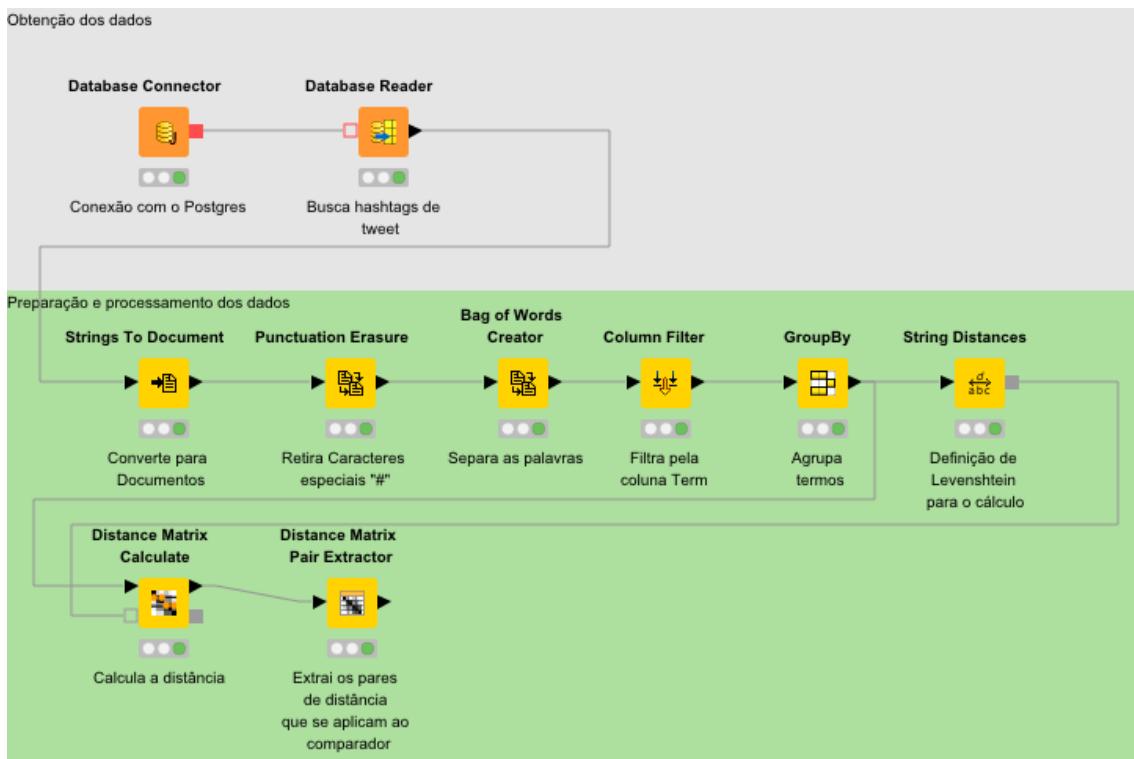


Figura 3.21: *Workflow* para geração de lista de termos próximos

3.3.3.1 Preparação e processamento dos dados

A etapa referente ao tratamento e processamento dos dados é exibida na figura 3.22. Os dados obtidos na seção 3.3.2 são formatados, distância entre termos calculada e resultados selecionados para exibição no gráfico.

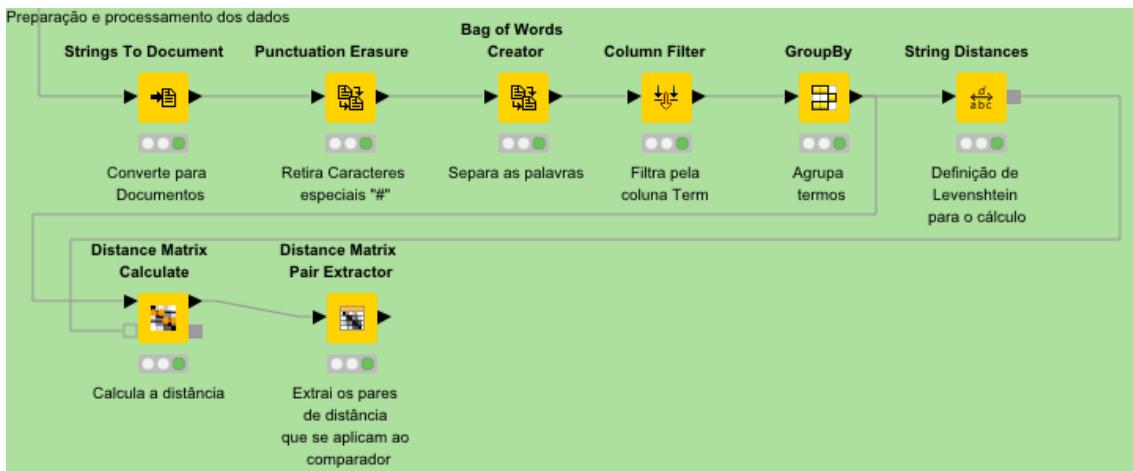


Figura 3.22: Etapa de preparação e processamento dos dados

O nó *String to Document* [40] converte uma *string* para documento. De cada registro obtido será criado um documento.

Na figura 3.23 podem ser vistas algumas configurações para criação desses documentos. Os campos título, texto e autor de cada documento preenche-se por dados do campo “*hashtags*”.

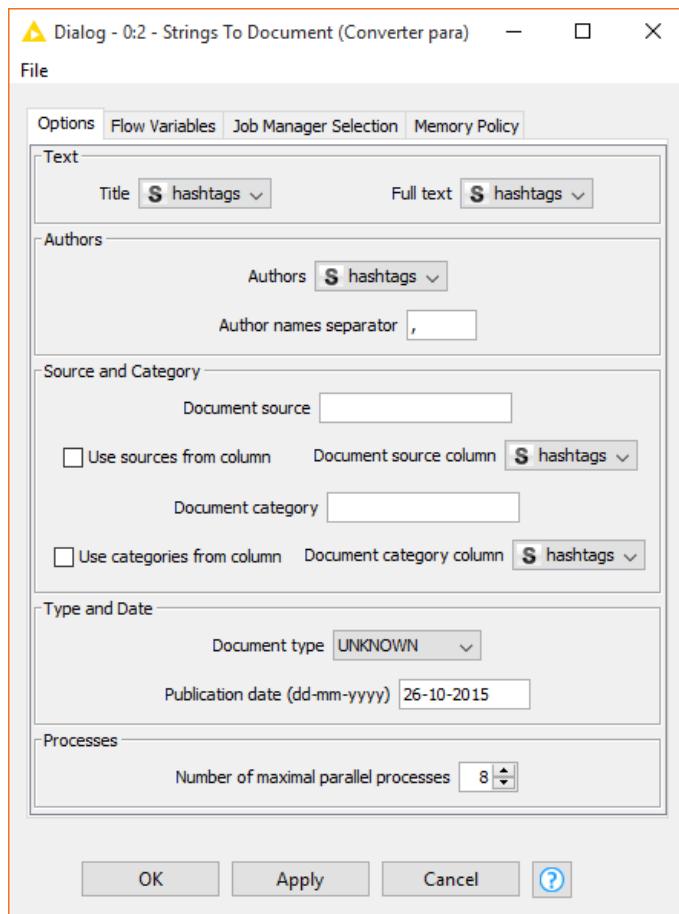


Figura 3.23: Configuração do nó *Strings To Document* no workflow de geração de lista de termos próximos

Na próxima etapa é utilizado o nó *Punctuation Erasure* [32], apresentado na figura 3.24, que pode ser usado para retirada de qualquer pontuação, no nosso caso, a cerquilha (“#”), presente nas *hashtags*.

Nas propriedades, podem ser configurados um pré-processamento profundo, o documento utilizado e se a coluna original também estará no resultado.

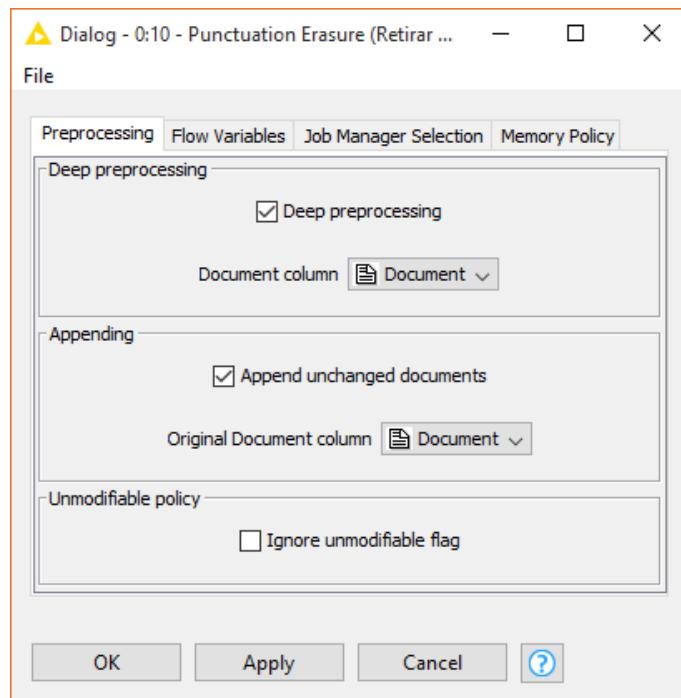


Figura 3.24: Configuração do nó *Punctuation Erasure* no workflow de geração de lista de termos próximos

O nó *Bag of Words Creator* [3], apresentado na figura 3.25, é utilizado para criar um repositório de termos de cada documento.

No exemplo, é selecionada a coluna que será utilizada para criação do repositório e o resultado obtido, apresentado na figura 3.26.

3.3. ESPECIFICAÇÃO DO WORKFLOW

31

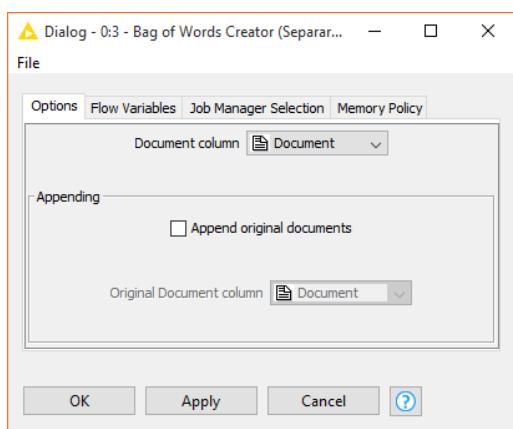


Figura 3.25: Configuração do nó *Bag of Words* no *workflow* de geração de lista de termos próximos

Documents output table - 0:3 - Bag of ...		
Properties		Flow Variables
Table "default" - Rows: 106279 Spec - Columns: 2		
Row ID	T Term	Document
Row1	foradilma	"foradilma foracunha"
Row2	foracunha	"foradilma foracunha"
Row3	foradilma	"foradilma foracunha"
Row4	foracunha	"foradilma foracunha"
Row5	foradilma	"foradilma foracunha"
Row6	foracunha	"foradilma foracunha"
Row7	foradilma	"foradilma foracunha"
Row8	foracunha	"foradilma foracunha"
Row9	foradilma	"foradilma foracunha"
Row10	foracunha	"foradilma foracunha"
Row11	foradilma	"foradilma foracunha"
Row12	foracunha	"foradilma foracunha"
Row13	naovaltergol...	"naovaltergolpe foracunha"
Row14	foracunha	"naovaltergolpe foracunha"
Row15	naovaltergol...	"naovaltergolpe foracunha"
Row16	foracunha	"naovaltergolpe foracunha"
Row17	foracunha	"foracunha"

Figura 3.26: Resultado da execução do nó *Bag of Words* no *workflow* de geração de lista de termos próximos

O nó *Column Filter* [6], apresentado na figura 3.27, é responsável pela seleção das colunas que serão utilizadas no próximo nó. No caso, foi selecionada a coluna “Term”, onde encontra-se as hashtags, apresentado na figura 3.28.

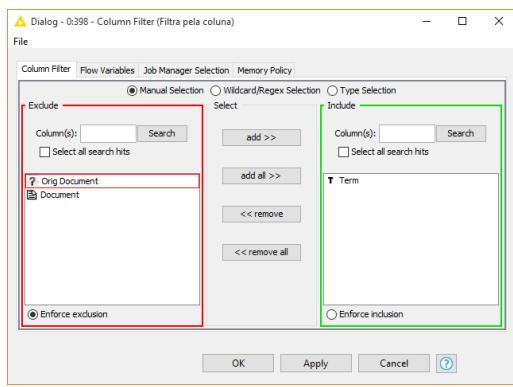


Figura 3.27: Configuração do nó *Column Filter* no *workflow* de geração de lista de termos próximos

Filtered table - 0:398 - Column Filter (Filtre pela coluna)		
Table "default" - Rows: 106279 Spec - Column: 1		Properties Flow Variables
Row ID	T Term	
Row1	foradilma	
Row2	foracunha	
Row3	foradilma	
Row4	foracunha	
Row5	foradilma	
Row6	foracunha	
Row7	foradilma	
Row8	foracunha	
Row9	foradilma	
Row10	foracunha	
Row11	foradilma	
Row12	foracunha	
Row13	naovaltergolpe	
Row14	foracunha	
Row15	naovaltergolpe	
Row16	foracunha	
Row17	foracunha	
Row18	foradilma	

Figura 3.28: Resultado da execução do nó *Column Filter* no *workflow* de geração de lista de termos próximos

Após essas etapas, é realizado um agrupamento dos termos idênticos através do

nó *GroupBy* [14], apresentado na figura 3.29. Neste, escolhem-se as colunas que serão utilizadas para o agrupamento.

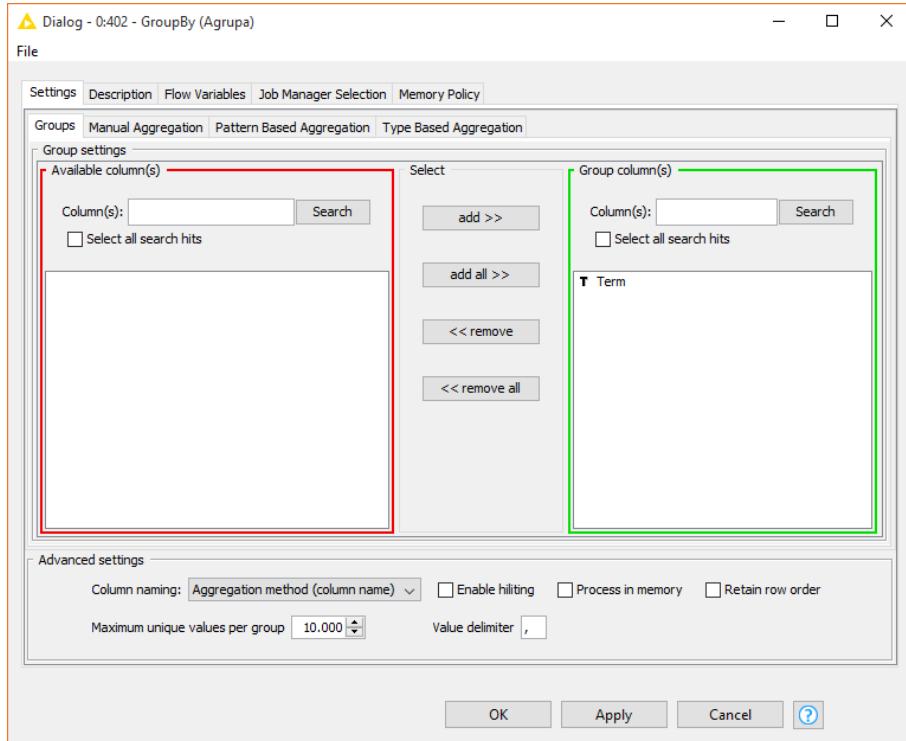


Figura 3.29: Configuração do nó *GroupBy* no *workflow* de geração de lista de termos próximos

Os nós *String Distances* [37], *Distance Matrix Calculate* [12], *Distance Matrix Pair Extractor* [31] realizam o cálculo da distância de Levenshtein [1, 61].

O nó *String Distances* [37] define o tipo de cálculo de distância utilizada sobre as *strings*, no caso, distância de Levenshtein [1, 61].

O nó *Distance Matrix Calculate* [12], apresentado na figura 3.30, cria a matriz de distância de “termos X termos”. Neste se escolhe o nome da coluna adicionada com as distâncias.

O nó *Distance Matrix Pair Extractor* [31], apresentado na figura 3.31, extrai cada par da matriz e o valor de sua distância. Neste, escolhe-se a coluna com os termos, com o vetor de distâncias e a condição para criação dos pares.

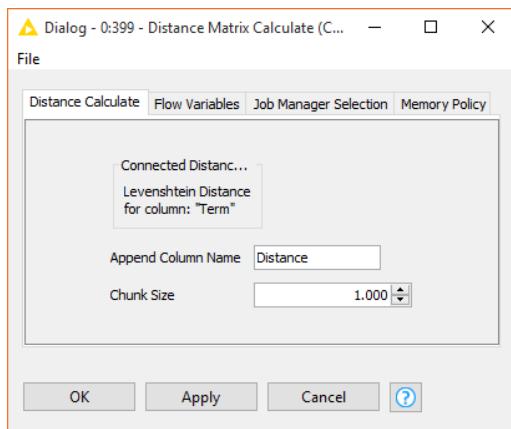


Figura 3.30: Configuração do nó *Distance Matrix Calculate* no *workflow* de geração de lista de termos próximos

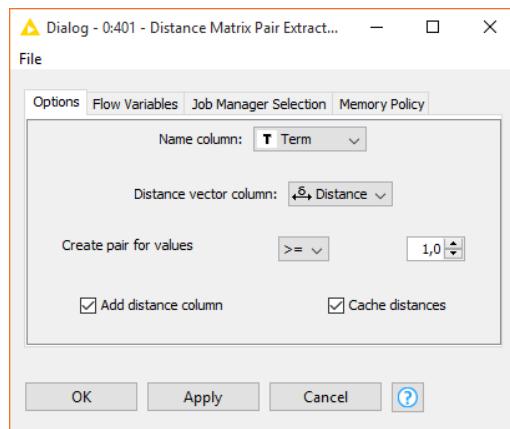


Figura 3.31: Configuração do nó *Distance Matrix Pair Extractor* no *workflow* de geração de lista de termos próximos

Utilizou-se a distância de Levenshtein [1, 61] no valor 1,0(um), por possibilitar o acerto de inúmeros erros básicos nos termos, como erros de digitação por parte do usuário, por exemplo.

O resultado do processamento é uma lista de pares de termos. Cada par foi analisado manualmente quanto a semântica em suas ocorrências no banco de dados, ocasionando uma tradução mais correta. Após esse processo, os termos da coluna “De” foram traduzidos na base de dados utilizando a coluna “Para”.

Abaixo pode ser visualizada, na tabela 3.1, alguns exemplos de tradução.

Tabela 3.1: Exemplos do dicionário

De	Para
<i>aocionever</i>	<i>aecionever</i>
<i>cadeualula</i>	<i>cadeialula</i>
<i>cartadotemmer</i>	<i>cartadotemer</i>
<i>ficardilma</i>	<i>ficadilma</i>
<i>foorapt</i>	<i>forapt</i>
<i>impechment</i>	<i>impeachment</i>
<i>impechmentja</i>	<i>impeachmentja</i>
<i>impsachmentja</i>	<i>impeachmentja</i>
<i>impitiman</i>	<i>impeachment</i>
<i>lullanacadeia</i>	<i>lulanacadeia</i>

3.3.4 Geração do gráfico de frequências de *hashtags*

Essa subseção apresenta em detalhes o *workflow* para visualização da frequência a qual as *hashtags* são mencionadas pelos usuários, apresentado na figura 3.32. Este gráfico possui o formato de barras por possibilitar a comparação entre as *hashtags*. Na figura, pode ser visto o *workflow* completo com as fases de obtenção dos dados a partir da base de dados, de preparação e processamento desses dados e da exibição do resultado.

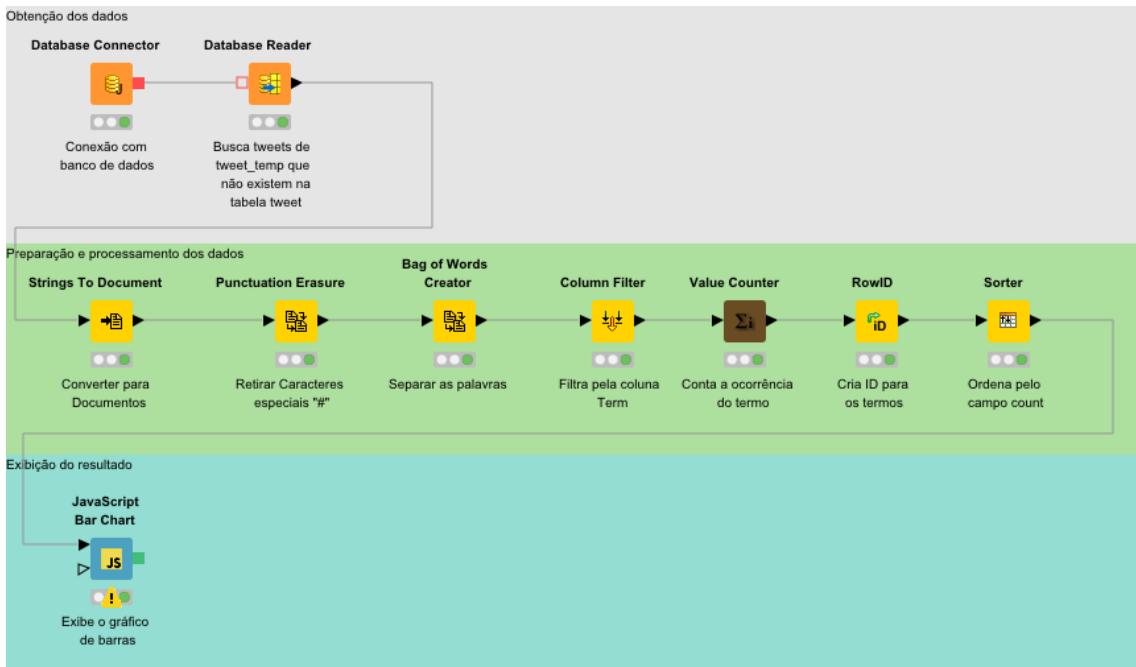


Figura 3.32: *Workflow* para geração do gráfico de frequências de hashtags

3.3.4.1 Preparação e processamento dos dados

Nessa subseção é apresentada a etapa do *workflow*, apresentado na figura 3.33, referente ao tratamento e processamento dos dados que possibilita a utilização na subseção posterior.

Os dados obtidos na seção 3.3.2 são formatados, frequência contabilizada e resultados ordenados para exibição no gráfico.

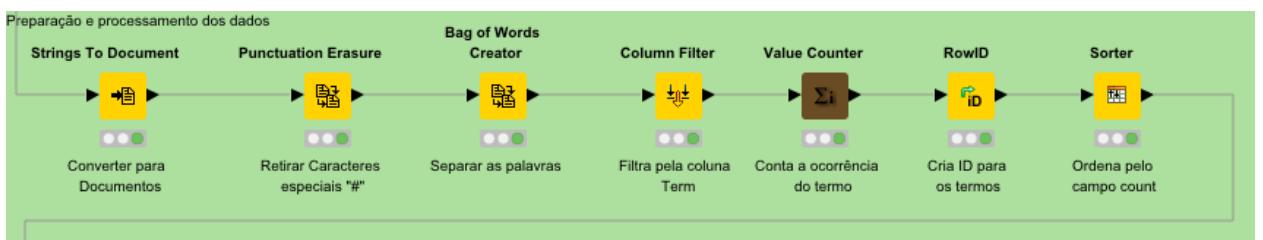


Figura 3.33: Etapa de preparação e processamento dos dados

As etapas de conversão de *strings* para documentos, retirada de cerquilha ("#"),

criação de repositório de termos de cada documento e seleção da coluna *Term*, são as idênticas as praticadas na seção 3.3.3.1.

O nó *Value Counter* [58], que pode ser visto na figura 3.34, apresenta a frequência da ocorrência de um valor na coluna. No caso, na coluna "Term", contabiliza a frequência de utilização das *hashtags* pelos usuários e tem seu resultado apresentado na figura 3.35.

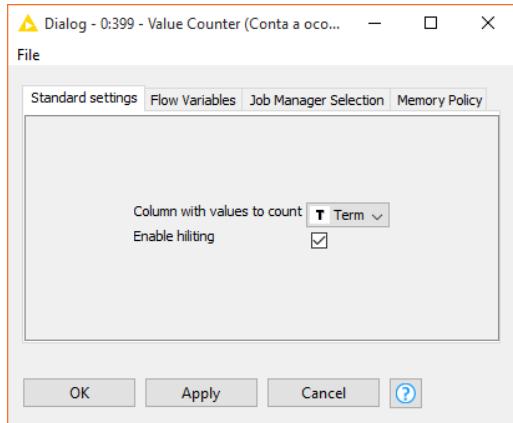


Figura 3.34: Configuração do nó *Value Counter* no *workflow* de geração do gráfico de frequências de hashtags

Occurrences - 0:3...		X
File		
Properties		Flow Variables
Table "default" - Rows: 3698	Spec - Column: 1	
Row ID	count	
foracunha[]	23753	
foradilma[]	23614	
forapt[]	8007	
naovaitergolpe[]	3998	
impeachment[]	2802	
lulanacademia[]	1973	
cunhanacademia[]	1909	
impeachmentdilma[]	1759	
dilmafica[]	1544	
impeachmentja[]	1498	
foralulada[]	1466	
povonarua13dez[]	1343	
vemprarua20dez[]	1111	
nataisemdilma[]	743	
brasil[]	696	
ficadilma[]	678	
delatadelcidio[]	655	

Figura 3.35: Resultado da execução do nó *Value Counter* no *workflow* de geração do gráfico de frequências de hashtags

O nó *RowID* [33], apresentado na figura 3.36, é responsável por criar um identificador para cada registro obtido do nó anterior. Ele apresenta duas opções para criação do identificador: substituir uma coluna existente pela coluna com o identificador ou criar uma nova coluna com o identificador. No *workflow* apresentado, foi criado uma nova coluna de nome “termo”.

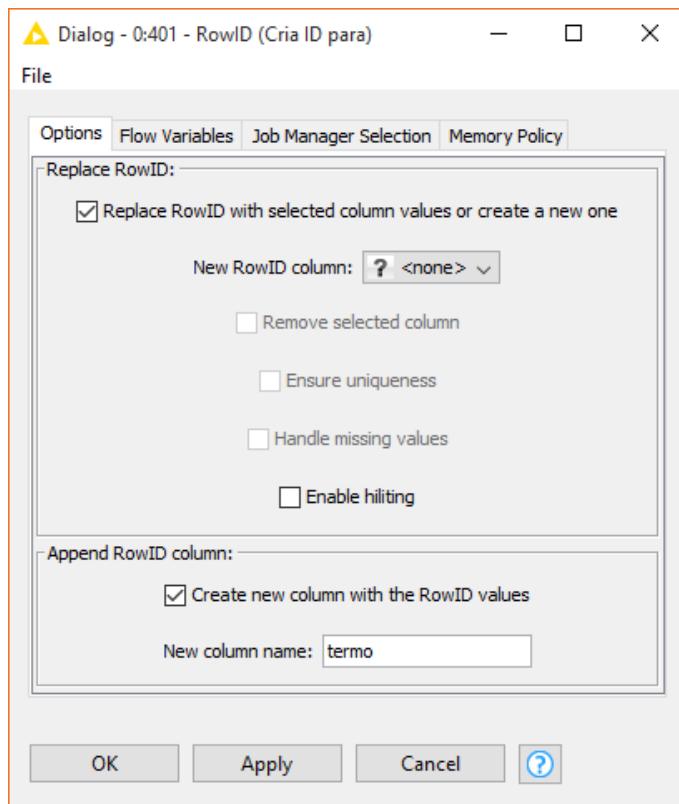


Figura 3.36: Configuração do nó *RowID* no *workflow* de geração do gráfico de frequências de hashtags

Para finalizar esta subseção, é utilizado o nó *Sorter* [36], apresentado na figura 3.37. Que tem como objetivo ordenar um campo selecionado. No caso, o campo *count* foi escolhido juntamente com uma ordenação decrescente.

3.3.4.2 Exibição do Resultado

Nesta subseção, o *workflow* é finalizado exibindo os resultados das subseções anteriores. O componente *JavaScript Bar Chart*, apresentado na figura 3.38, é utilizado a fim de criar um gráfico de barras para exibição das frequências de ocorrência das *hashtags*.

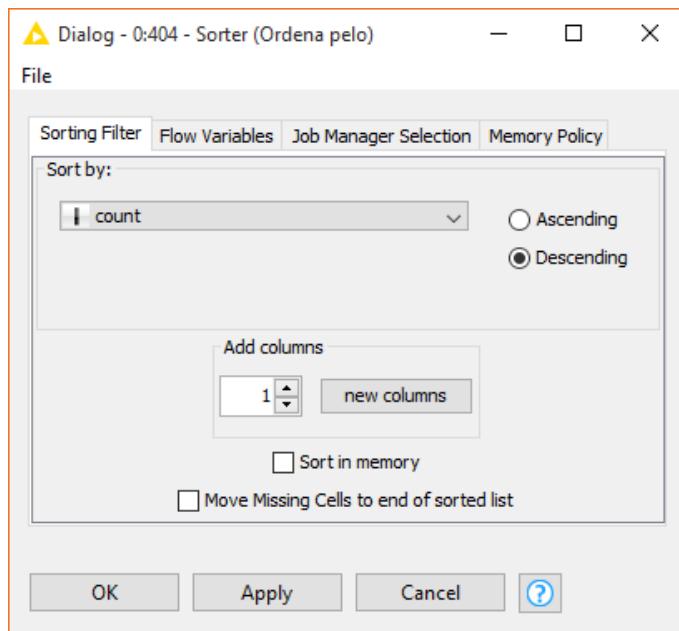


Figura 3.37: Configuração do nó *Sorter* no *workflow* de geração do gráfico de frequências de hashtags

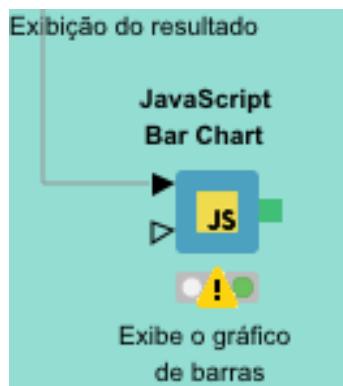


Figura 3.38: Etapa de exibição do resultado do *workflow* de geração do gráfico de frequências de hashtags

Nas configurações apresentadas nas figuras 3.39 e 3.40, são selecionados a coluna com as frequências e a coluna com as categorias a serem exibidas, o método de agregação, o título, a orientação das barras, e o tamanho da imagem em *pixels*.

No *workflow*, foram escolhidos a coluna *count* para a frequência, termo para a categoria, *Average* para método de agregação, título “Gráfico de frequências” e tamanho de imagem 1000 x 600px.

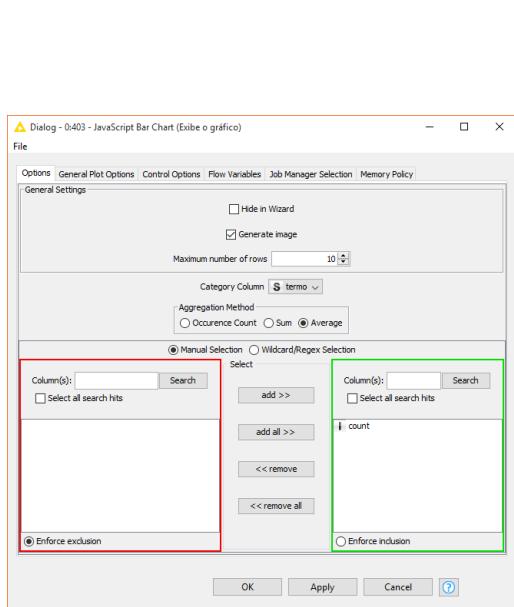


Figura 3.39: Seção de configuração de opções do nó *JavaScript Bar Chart* para geração da lista de frequência

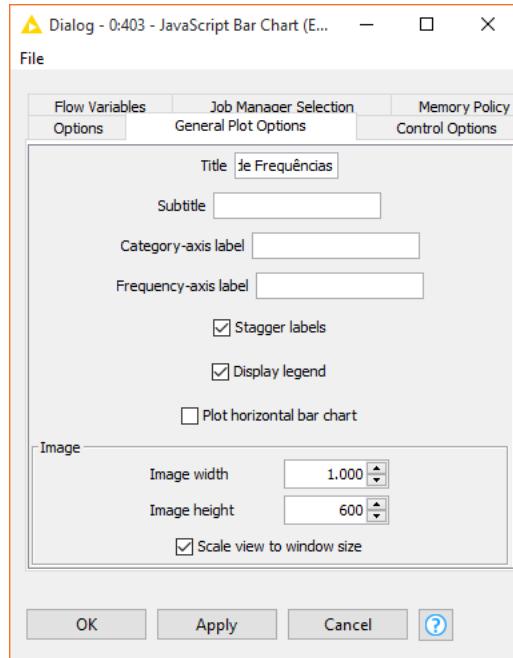


Figura 3.40: Seção de configuração de opções de plotagem do nó *JavaScript Bar Chart* para geração da lista de frequência

3.3.5 Geração da lista de frequência

Com base na lista de termos criados na seção 3.3.4, na etapa de preparação e processamento dos dados, para visualização em gráfico, foi realizada uma modificação na base de dados para que somente 20% dos termos de maior frequência fossem utilizados nas etapas de geração da matriz de coocorrência, seção 3.3.6, e aplicação da regra de associação [62, 60], seção 3.3.7.

O valor de 20% foi utilizado com base no princípio de Pareto [34, 30]. Tal princípio foi proposto por Joseph M. Juran e nomeado em homenagem ao economista Vilfredo Pareto. De acordo com este princípio, 80% das consequências decorrem de 20% das causas.

Durante pesquisas, Pareto descobriu que 80% da riqueza estava nas mãos de apenas 20% da população.

Através desta lei é possível afirmar que mais de 80% das *hashtags* utilizadas estão em 20% do total de *hashtags*.

3.3.6 Matriz de coocorrência

Esta subseção objetiva definir e ilustrar a criação do *workflow*, apresentado na figura 3.41, que possibilita a visualização do grau de ocorrência entre duas *hashtags*. Essa visualização é possibilitada por um mapa de calor onde as faixas mais escuras em tom de vermelho são criadas a partir da maior coocorrência entre dois termos.

O *workflow* apresentado na figura 3.41, exibe as fases de obtenção dos dados, de preparação e processamento desses dados e da exibição do resultado.

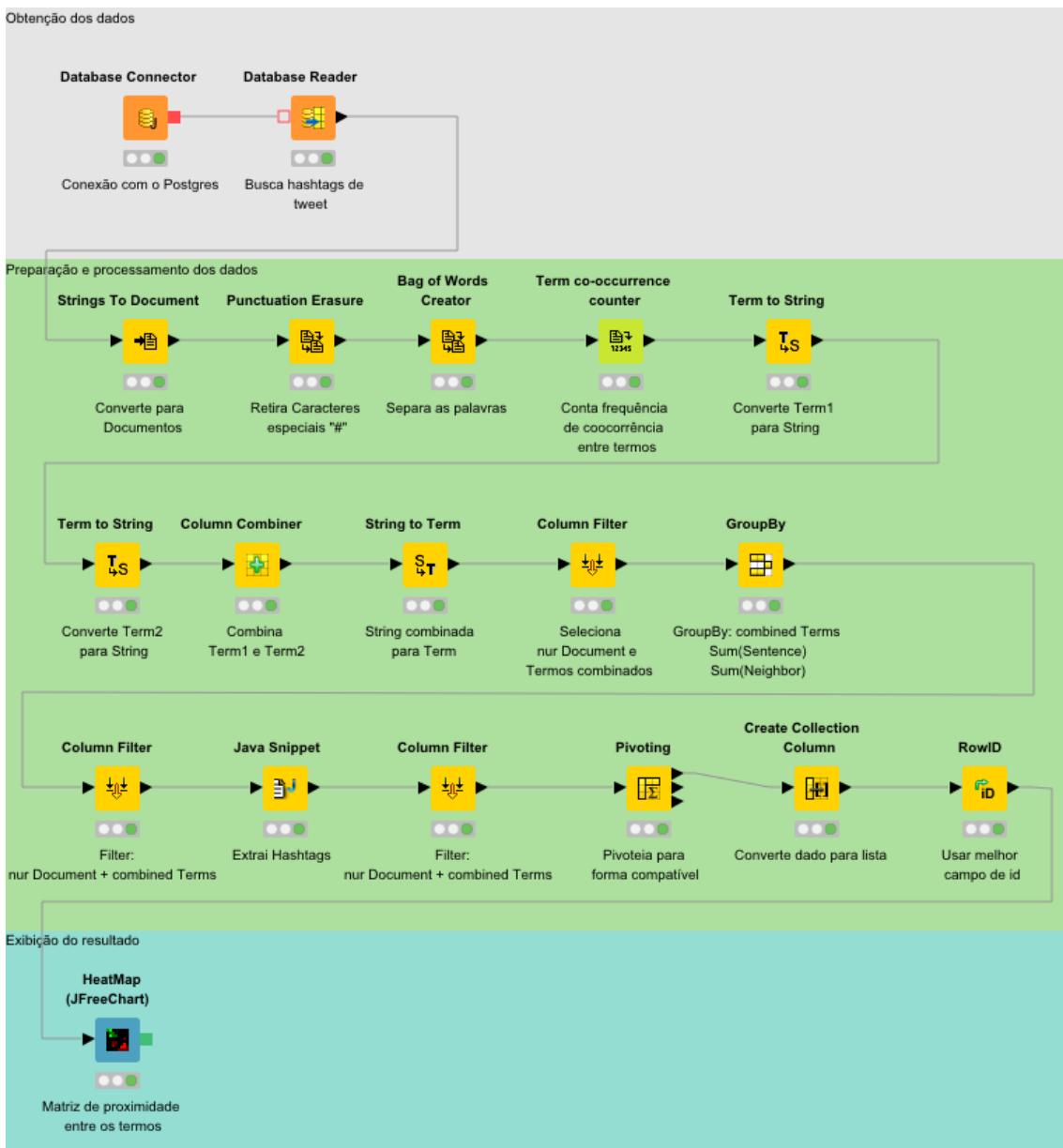


Figura 3.41: Workflow da geração da Matriz de coocorrência

3.3.6.1 Preparação e processamento dos dados

Esta etapa, referente ao tratamento e processamento dos dados, exibe como os dados são tratados com a finalidade da utilização na etapa de exibição do resultado por um mapa de calor, como pode ser visto na figura 3.42. O objetivo principal é processar os dados selecionando os termos coocorrentes. Os dados obtidos na seção 3.3.2 são formatados, frequência de coocorrência é extraída e os resultados são

agrupados e adaptados para as necessidades da próxima etapa.



Figura 3.42: Etapa de preparação e processamento dos dados

As etapas de conversão de *strings* para documentos, retirada de cerquilha ("#"), criação de repositório de termos de cada documento, são idênticas as praticadas na seção 3.3.3.1.

O nó *Term co-occurrence counter* [44], apresentado na figura 3.43, conta o número de coocorrências entre os termos selecionados em determinado documento. As entradas são: a coluna de documentos e a de termos. No caso, para contagem de coocorrência entre termos, deve ser utilizada a opção “*Sentence*” em *Co-occurrence level*. O resultado pode ser observado na figura 3.44.

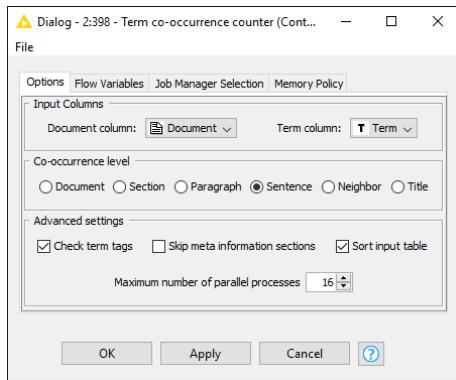


Figura 3.43: Configuração do nó *Term co-occurrence counter* no workflow de geração da Matriz de coocorrência

Co-occurrence table - 0:398 - Term co-occurrence counter (Conta frequênci)				
File				
Table "default" - Rows: 40800 Spec - Columns: 6 Properties Flow Variables				
Row ID	Document	Term1	Term2	Sentence cooccurrence
Row14791	"foracunha naovaitergolpe foracunha naovaitergolpe foracu...	foracunha[]	naovaitergolpe[]	10
Row14996	"foradilma foracunha foradilma foracunha foradilma foracun...	foracunha[]	foradilma[]	6
Row9655	"eusemprequisiter 1dhistory segundateindetremurasdv fora...	foradilma[]	ocupabrasilia[]	4
Row0	"1312 povonarua13dez vamojunto foradilma foracunha fora...	1312[]	povonarua13de...	2
Row1	"1312 povonarua13dez vamojunto foradilma foracunha fora...	povonarua1...	vamojunto[]	2
Row2	"1312 povonarua13dez vamojunto foradilma foracunha fora...	foradilma[]	vamojunto[]	2
Row3	"1312 povonarua13dez vamojunto foradilma foracunha fora...	foracunha[]	foradilma[]	2
Row4	"1312 povonarua13dez vamojunto foradilma foracunha fora...	foracunha[]	forapt[]	2
Row5	"1312 povonarua13dez vamojunto foradilma foracunha fora...	1312[]	vamojunto[]	2
Row6	"1312 povonarua13dez vamojunto foradilma foracunha fora...	1312[]	foradilma[]	2
Row7	"1312 povonarua13dez vamojunto foradilma foracunha fora...	1312[]	foracunha[]	2
Row8	"1312 povonarua13dez vamojunto foradilma foracunha fora...	1312[]	forapt[]	2
Row9	"1312 povonarua13dez vamojunto foradilma foracunha fora...	foradilma[]	povonarua13de...	2
Row10	"1312 povonarua13dez vamojunto foradilma foracunha fora...	foracunha[]	povonarua13de...	2
Row11	"1312 povonarua13dez vamojunto foradilma foracunha fora...	forapt[]	povonarua13de...	2
Row12	"1312 povonarua13dez vamojunto foradilma foracunha fora...	foracunha[]	vamojunto[]	2
Row13	"1312 povonarua13dez vamojunto foradilma foracunha fora...	forapt[]	vamojunto[]	2

Figura 3.44: Resultado da execução do nó *Term co-occurrence counter* no workflow de geração da Matriz de coocorrência

Os nós *Term to String* [45], apresentado na figura 3.45, *Column Combiner* [5], apresentado na figura 3.47 e *String to Term* [39], apresentado na figura 3.46 em sequência, tem com o objetivo concatenar as colunas “*Term1*” e “*Term2*” em somente uma coluna. Para tal, as colunas com os termos são transformadas em colunas do tipo *string* com o uso do nó *Term to String* [45]. Para a concatenação, utiliza-se do nó *Column Combiner* [5] junto as colunas do passo anterior. O resultado

da combinação pode ser visto na figura 3.48. Finalizando, a coluna resultante é transformada no tipo *Term* para utilização nos nós posteriores.

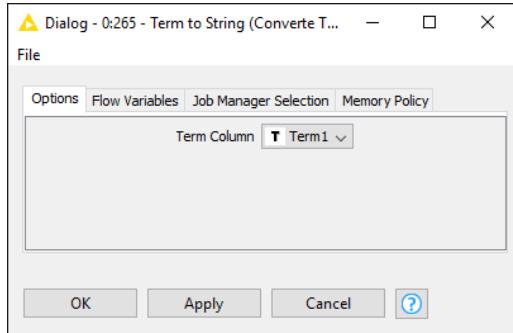


Figura 3.45: Configuração do nó *Term to String* no workflow de geração da Matriz de coocorrência

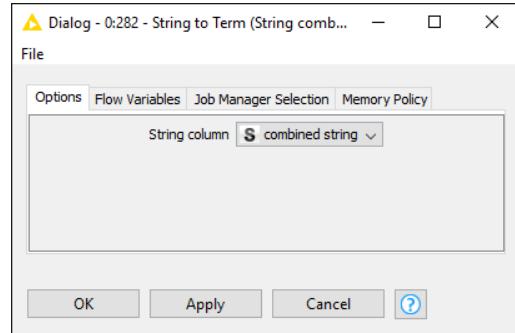


Figura 3.46: Configuração do nó *String to Term* no workflow de geração da Matriz de coocorrência

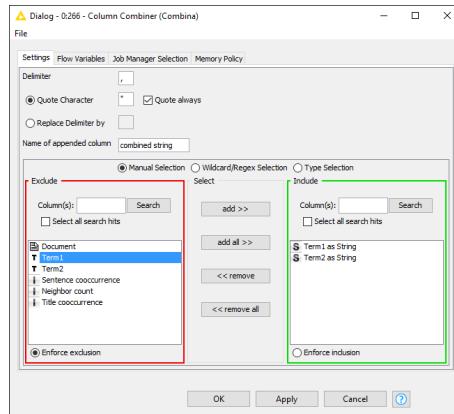


Figura 3.47: Configuração do nó *Column Combiner* no workflow de geração da Matriz de coocorrência

Row ID	Term1 as String	Term2 as String	combined string	Term
Row0	1312	povonarua13...	"1312","povonarua13dez"	"1312","povonarua13dez"
Row1	povonarua1...	vamojunto	"povonarua13dez","vamojunto"	"povonarua13dez","vamojunto"
Row2	foradilma	vamojunto	"foradilma","vamojunto"	"foradilma","vamojunto"
Row3	foracunha	foradilma	"foracunha","foradilma"	"foracunha","foradilma"
Row4	foracunha	forapt	"foracunha","forapt"	"foracunha","forapt"
Row5	1312	vamojunto	"1312","vamojunto"	"1312","vamojunto"
Row6	1312	foradilma	"1312","foradilma"	"1312","foradilma"
Row7	1312	foracunha	"1312","foracunha"	"1312","foracunha"
Row8	1312	forapt	"1312","forapt"	"1312","forapt"
Row9	foradilma	povonarua13...	"foradilma","povonarua13dez"	"foradilma","povonarua13dez"
Row10	foracunha	povonarua13...	"foracunha","povonarua13dez"	"foracunha","povonarua13dez"
Row11	forapt	povonarua13...	"forapt","povonarua13dez"	"forapt","povonarua13dez"
Row12	foracunha	vamojunto	"foracunha","vamojunto"	"foracunha","vamojunto"
Row13	forapt	vamojunto	"forapt","vamojunto"	"forapt","vamojunto"
Row14	foradilma	forapt	"foradilma","forapt"	"foradilma","forapt"
Row15	13confirma	foradilma	"13confirma","foradilma"	"13confirma","foradilma"
Row16	13beijos	foracunha	"13beijos","foracunha"	"13beijos","foracunha"

Figura 3.48: Resultado da execução do nó *String to Term* no workflow de geração da Matriz de coocorrência

O nó *Column Filter* [6], apresentado na figura 3.49, que é o responsável pela seleção das colunas que serão utilizadas no próximo nó, foi utilizado para a seleção das colunas *Document*, *Neighbor Count*, *Title cooccurrence* e *Term*.

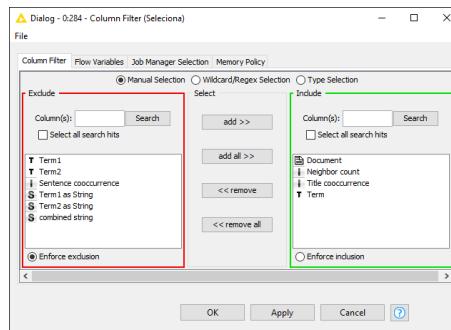


Figura 3.49: Configuração do nó *Column Filter* no workflow de geração da Matriz de coocorrência

O nó *GroupBy* [14] agrupa os termos de uma determinada coluna e outra determinada coluna é agregada conforme uma configuração específica de agregação.

No caso, a coluna de termos é a *Combined Terms*, apresentado na figura 3.50, onde temos os termos combinados. A configuração de agregação é para a soma dos valores obtidos na coluna *Title cooccurrence*, apresentado na figura 3.51, onde visualizamos a frequência de coocorrência de dois termos em cada documento. O objetivo da utilização desse nó é obter a frequência de coocorrência total entre os pares de termos, conforme pode ser visto na figura 3.52.

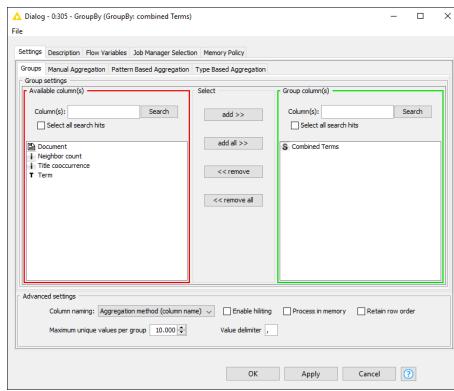


Figura 3.50: Configuração do nó *GroupBy* no *workflow* de geração da Matriz de coocorrência

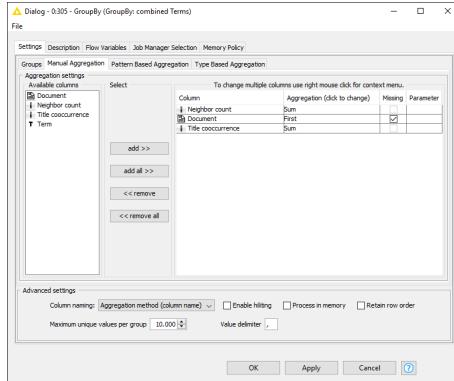


Figura 3.51: Seção *Manual Aggregation* da configuração do nó *GroupBy* no *workflow* de geração da Matriz de coocorrência

Group table - 0:305 - GroupBy (GroupBy: combined Terms)				
File				
Table "default" - Rows: 99 Spec - Columns: 4 Properties Flow Variables				
Row ID	Combined Terms	Sum(Next Row ID)	First(Document)	Sum>Title cooccurrence
Row14	"cunhanacadeia", "foracunha"	50	"cunhanacadeia foracunha"	26
Row49	"foracunha", "naovaitergolpe"	12	"foracunha cunhanacadeia ficadilma naovai..."	24
Row16	"cunhanacadeia", "naovaitergolpe"	2	"foracunha cunhanacadeia ficadilma naovai..."	16
Row12	"cunhanacadeia", "ficadilma"	30	"foracunha cunhanacadeia ficadilma naovai..."	15
Row32	"ficadilma", "foracunha"	0	"foracunha cunhanacadeia ficadilma naovai..."	15
Row33	"ficadilma", "naovaitergolpe"	30	"foracunha cunhanacadeia ficadilma naovai..."	15
Row66	"foradilma", "lulanacadeia"	22	"foradilma lulanacadeia"	12
Row70	"foradilma", "povonarua13dez"	22	"vazadilma13dez povonarua13dez foradilma"	11
Row73	"foradilma", "vazadilma13dez"	0	"vazadilma13dez povonarua13dez foradilma"	11
Row98	"povonarua13dez", "vazadilma13...	22	"vazadilma13dez povonarua13dez foradilma"	11
Row43	"foracunha", "golpenunciamais"	18	"foracunha golpenunciamais"	9
Row44	"foracunha", "golpeparlamentar"	10	"foracunha golpeparlamentar"	6
Row23	"dilmatica", "foracunha"	4	"foracunha cunhanacadeia dilmatica golpe..."	4
Row13	"cunhanacadeia", "forac"	6	"cunhanacadeia forac"	3
Row60	"foradilma", "forapt"	4	"foradilma foralula forapt vemprarua impeac"	3
Row64	"foradilma", "impeachment"	4	"impeachment foradilma"	3
Row8	"conselhode", "foracunha"	4	"foracunha conselhode tica"	2
Row9	"conselhode", "tica"	4	"foracunha conselhode tica"	2

Figura 3.52: Resultado da execução do nó *GroupBy* no *workflow* de geração da Matriz de coocorrência

Após obtidas as coocorrências entre os termos, os dados são formatados para utilização no nó de exibição da próxima etapa.

O nó *Column Filter* [6] seleciona as colunas *Combined Terms* e *Sum>Title cooccurrence*, apresentado na figura 3.53 e com resultado mostrado na figura 3.54.

O nó *Java Snippet* [23], apresentado na figura 3.55, é utilizado para separação dos termos em duas colunas, “Termo1” e “Termo2” e tem seu resultado mostrado na figura 3.56.

O nó *Column Filter* [6] seleciona somente as colunas *Sum>Title cooccurrence*, Termo1 e Termo2, eliminando a coluna *Combined Terms*(termos combinados), apresentado na figura 3.57 e com resultado mostrado na figura 3.58.

3.3. ESPECIFICAÇÃO DO *WORKFLOW*

48

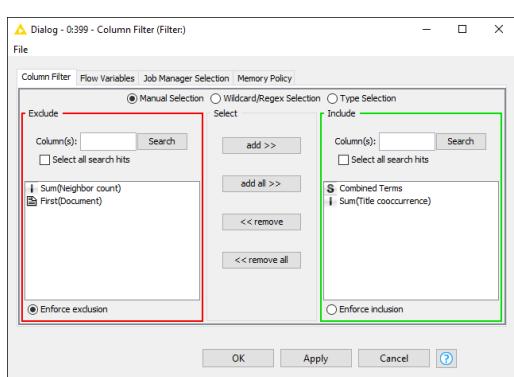


Figura 3.53: Configuração do nó *Column Filter* no workflow de geração da Matriz de co-
corrência

Filtered table - 0:399 - Column Filter (Filter:)

File

Table "default" - Rows: 99 Spec - Columns: 2 Properties Flow Variables

Row ID	Combined Terms	Sum(Title cooccurrence)
Row0	"aceitem", "dilmatica"	1
Row1	"aceitem", "foracunha"	1
Row2	"aceitem", "lei"	1
Row3	"aceitem", "naotevegolpe"	1
Row4	"bondia", "foracunha"	1
Row5	"bondia", "vivaademocracia"	1
Row6	"cadeomovimentoestudantil..."	1
Row7	"cadeomovimentostudantil..."	1
Row8	"conselhode", "foracunha"	2
Row9	"conselhode", "tica"	2
Row10	"cunha", "foracunha"	2
Row11	"cunhanacademia", "dilmatica"	2
Row12	"cunhanacademia", "ficadilma"	15
Row13	"cunhanacademia", "forac"	3
Row14	"cunhanacademia", "foracunha"	26
Row15	"cunhanacademia", "golpeparla... 1	1
Row16	"cunhanacademia", "naoaverta... 16	16
Row17	"cunhanacademia", "policiade... 1	1

Figura 3.54: Resultado da execução do nó *Column Filter* no *workflow* de geração da Matriz de coocorrência

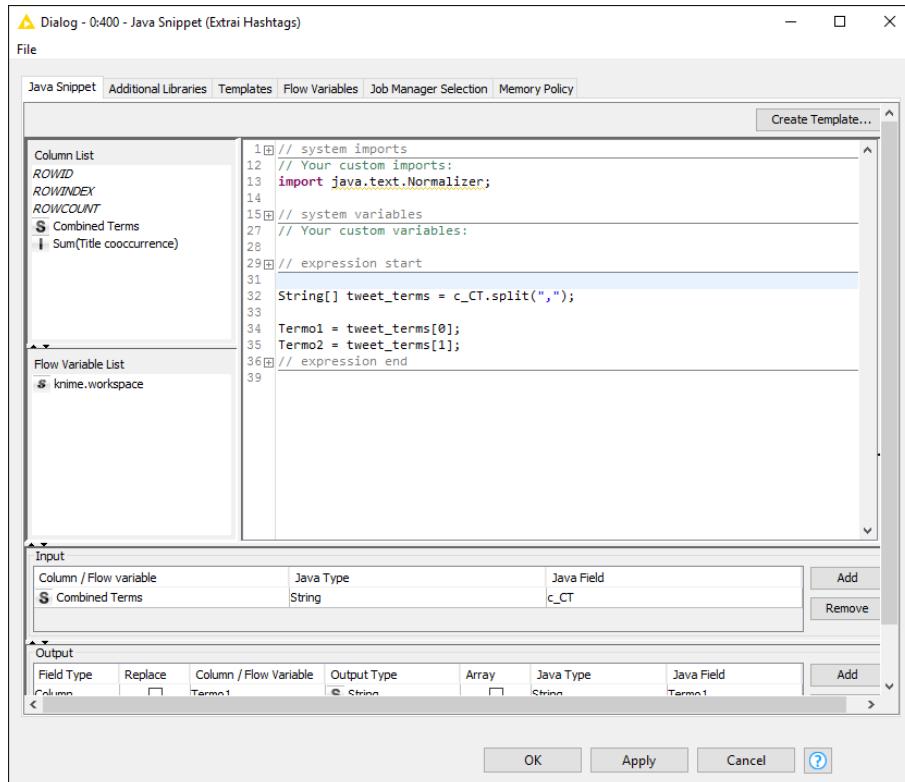


Figura 3.55: Configuração do nó *Java Snippet* no *workflow* de geração da Matriz de coocorrência

Row ID	Combined Terms	Su...	Termo1	Termo2
Row14	"cunhanacadeia",...	26	"cunhanaca..."	"foracunha"
Row49	"foracunha", "nao...	24	"foracunha"	"naovaiterg..."
Row16	"cunhanacadeia",...	16	"cunhanaca..."	"naovaiterg..."
Row12	"cunhanacadeia",...	15	"cunhanaca..."	"ficadilma"
Row32	"ficadilma", "forac...	15	"ficadilma"	"foracunha"
Row33	"ficadilma", "naov...	15	"ficadilma"	"naovaiterg..."
Row66	"foradilma", "ulan...	12	"foradilma"	"ulanacadeia"
Row70	"foradilma", "povo...	11	"foradilma"	"povonarua..."
Row73	"foradilma", "vaza...	11	"foradilma"	"vazadilma1..."
Row98	"povonarua13dez..."	11	"povonarua..."	"vazadilma1..."
Row43	"foracunha", "glop...	9	"foracunha"	"golpenuncia..."
Row44	"foracunha", "glop...	6	"foracunha"	"golpeparla..."
Row23	"dilmafica", "forac...	4	"dilmafica"	"foracunha"
Row13	"cunhanacadeia",...	3	"cunhanaca..."	"forac"
Row60	"foradilma", "forapt"	3	"foradilma"	"forapt"
Row64	"foradilma", "impe...	3	"foradilma"	"impeachment"
Row8	"conselhode", "for...	2	"conselhode"	"foracunha"
Row9	"conselhode", "tica"	2	"conselhode"	"tica"

Figura 3.56: Resultado da execução do nó *Java Snippet* no workflow de geração da Matriz de coocorrência

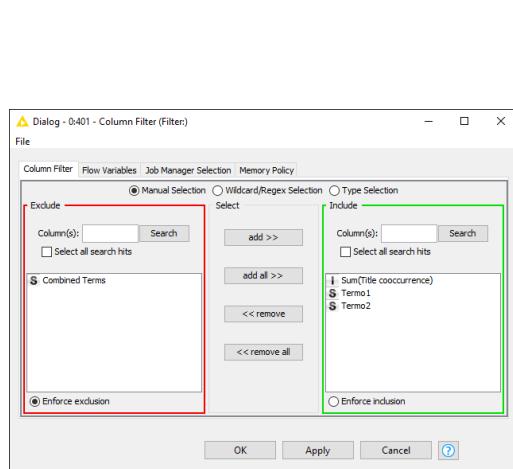


Figura 3.57: Configuração do nó *Column Filter* no workflow de geração da Matriz de coocorrência

Row ID	Sum(Tit...)	Termo1	Termo2
Row0	1	"aceitem"	"dilmafica"
Row1	1	"aceitem"	"foracunha"
Row2	1	"aceitem"	"lei"
Row3	1	"aceitem"	"naotevegol..."
Row4	1	"bomdia"	"foracunha"
Row5	1	"bomdia"	"vivaademo..."
Row6	1	"cadeomovi..."	"cunhanaca..."
Row7	1	"cadeomovi..."	"foracunha"
Row8	2	"conselhode"	"foracunha"
Row9	2	"conselhode"	"tica"
Row10	2	"cunha"	"foracunha"
Row11	2	"cunhanaca..."	"dilmafica"
Row12	15	"cunhanaca..."	"ficadilma"
Row13	3	"cunhanaca..."	"forac"
Row14	26	"cunhanaca..."	"foracunha"
Row15	1	"cunhanaca..."	"golpeparla..."
Row16	16	"cunhanaca..."	"naovaiterg..."

Figura 3.58: Resultado da execução do nó *Column Filter* no workflow de geração da Matriz de coocorrência

O uso do *Pivoting* [48], apresentado na figura 3.59, cria uma matriz de “termos X termos”, onde cada termo é comparado com todos os outros termos utilizando suas frequências de coocorrência. Esse nó é utilizado para dar o formato necessário ao gráfico, como pode ser visto na figura 3.60.

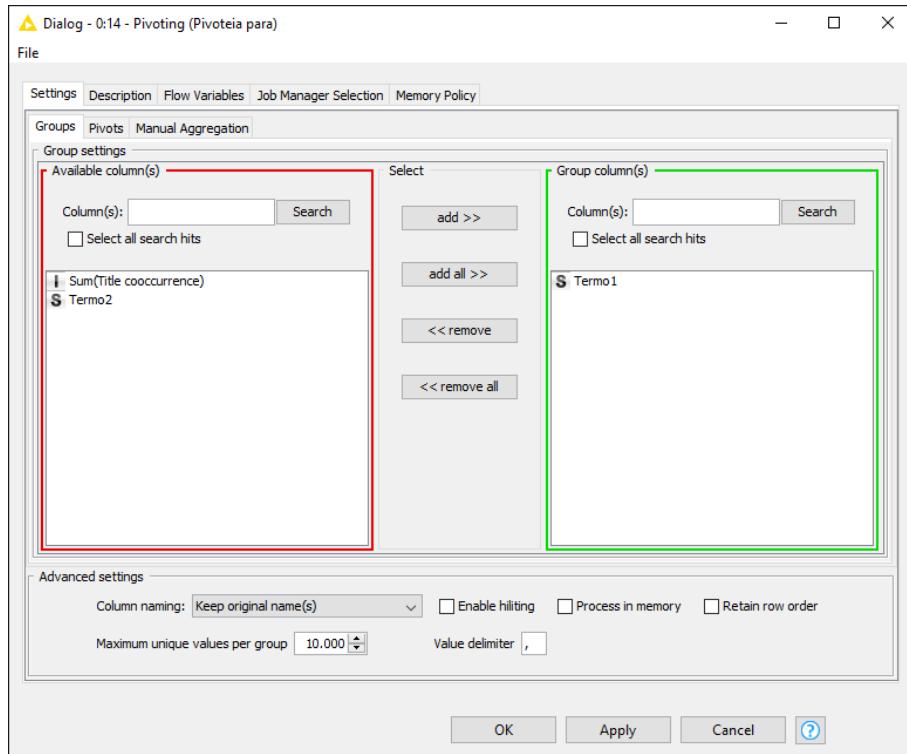


Figura 3.59: Configuração do nó *Pivoting* no *workflow* de geração da Matriz de coocorrência

Table "default" - Rows: 6 Spec - Columns: 10 Properties Flow Variables											
Row ID	Termo1	"fadicl..."	"forac..."	"foracu..."	"naoval..."	"forare..."	"natais..."	"forapt..."	"impeac..."	"ulanac..."	
Row0	"cunhanacad..."	1	1	2	1	?	?	?	?	?	
Row1	"fadiclimes"	?	?	1	1	?	?	?	?	?	
Row2	"foracurinha"	?	?	?	2	1	1	?	?	?	
Row3	"foradilma"	?	?	?	?	?	?	1	1	4	
Row4	"forapt"	?	?	?	?	?	?	?	1	1	
Row5	"impeachment"	?	?	?	?	?	?	?	?	1	

Figura 3.60: Resultado da execução do nó *Pivoting* no *workflow* de geração da Matriz de coocorrência

O nó *Create Collection Column* [7], apresentado na figura 3.61, cria uma lista por linha com as colunas selecionadas na configuração. No *workflow*, foram utilizadas as colunas com frequências de coocorrência e adquiridas listas com estas.

3.3. ESPECIFICAÇÃO DO WORKFLOW

51

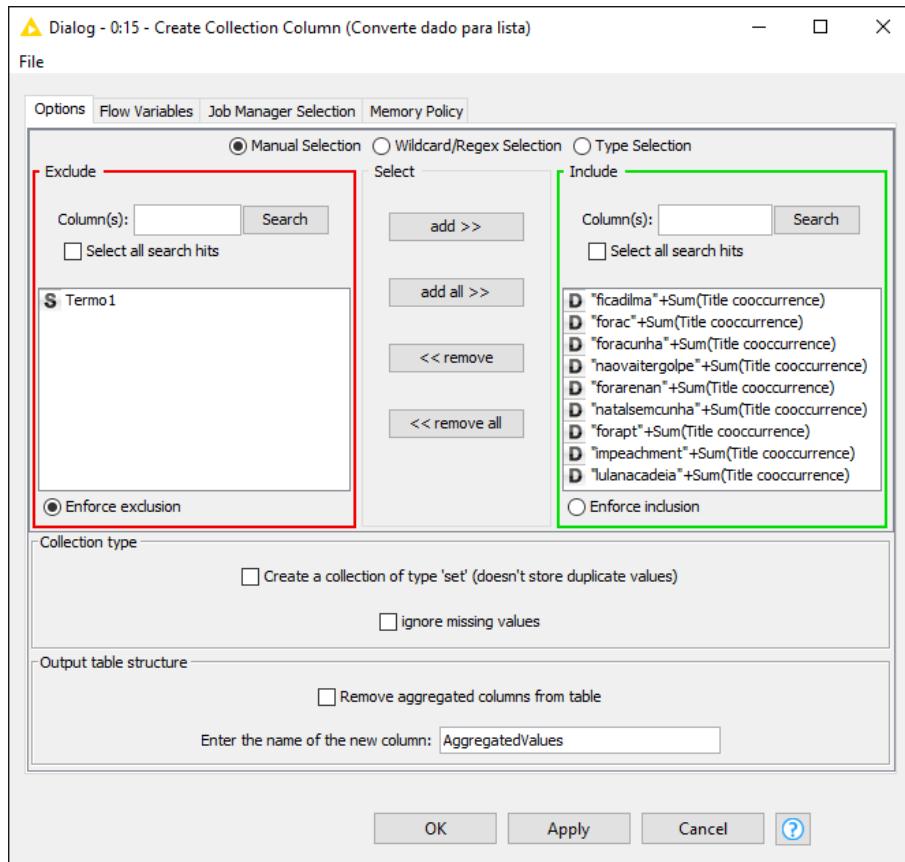


Figura 3.61: Configuração do nó *Create Collection Column* no workflow de geração da Matriz de coocorrência

Input data with new collection column - 0:15 - Create Collection Column (Converte dado para lista)											
Table "default" - Rows: 6 Spec - Columns: 11 Properties Flow Variables											
Row ID	Termo1	ficadilma	forac	foracunha	naovai... tergolpe	forare... naturais	natais...	forapt...	impeach...	lulan...	(...) AggregatedValues
Row0	cunhanacad...	1	1	2	1	?	?	?	?	?	[1,0,1,0,2,0,...]
Row1	"ficadilma"	?	?	1	1	?	?	?	?	?	[?,?,1,0,...]
Row2	"foracunha"	?	?	?	2	1	1	?	?	?	[?,?,?,?,...]
Row3	"foradilma"	?	?	?	?	?	?	1	1	4	[?,?,?,?,...]
Row4	"forapt"	?	?	?	?	?	?	?	1	1	[?,?,?,?,...]
Row5	"impeachment"	?	?	?	?	?	?	?	?	1	[?,?,?,?,...]

Figura 3.62: Resultado da execução do nó *Create Collection Column* no workflow de geração da Matriz de coocorrência

O nó *RowID* [33], apresentado na figura 3.63, é responsável por criar um identificador para cada registro obtido do nó anterior. No caso, cada termo tornou-se um identificador.

Processed data - 0:17 - RowID (Usar melhor)										
File										
Table "default" - Rows: 6 Spec - Columns: 11 Properties Flow Variables										
Row ID	S Termo 1	D "fadicad... 1	D "forac... 2	D "foracu... 1	D "naoval... ?	D "forare... ?	D "natais... ?	D "forapt... ?	D "impeac... ?	D "ulanac... [1,0,1,0,2,0,...]
"cunhanacad...	"cunhanacad...	1	1	2	1	?	?	?	?	[1,0,1,0,2,0,...]
"fadiclma"	"fadiclma"	?	?	1	1	?	?	?	?	[?,?,1,0,...]
"foracuinha"	"foracuinha"	?	?	?	2	1	1	?	?	[?,?,?,1,...]
"foradima"	"foradima"	?	?	?	?	?	?	1	1	[?,?,?,1,...]
"forapt"	"forapt"	?	?	?	?	?	?	?	1	[?,?,?,1,...]
"impeachment"	"impeachment"	?	?	?	?	?	?	?	1	[?,?,?,1,...]

Figura 3.63: Resultado da execução do nó *RowID* no *workflow* de geração da Matriz de coocorrência

3.3.6.2 Exibição do resultado

Nesta subseção, o *workflow* é finalizado exibindo os resultados das subseções anteriores. O componente *HeatMap (JFreeChart)* [21], apresentado na figura 3.64, é utilizado com objetivo de criar uma matriz de calor para exibição da coocorrência entre os termos das *hashtags*.

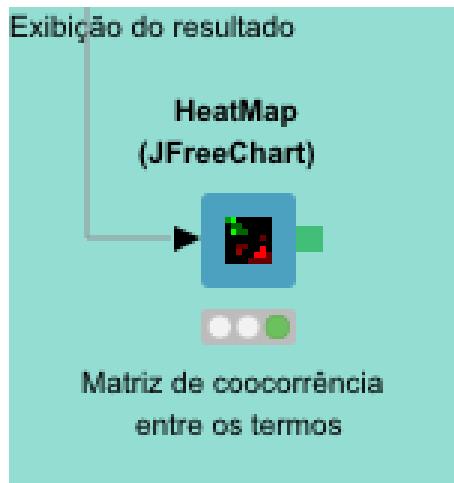


Figura 3.64: Etapa de exibição da matriz de coocorrência no *workflow* de geração da Matriz de coocorrência

Nas configurações, apresentadas nas figuras 3.65, 3.66 e 3.67, são selecionadas as colunas utilizadas para preencher cada categoria dos eixos, número de linhas exibidas, cores e legenda para escala, orientação das categorias e tamanho da imagem em *pixels*.

No *workflow*, foram escolhidas as colunas adquiridas no processo da subseção anterior, que todas as linhas sejam exibidas, cores vermelho e verde para escala,

orientação vertical para o eixo X e horizontal para o eixo Y e geração de imagem de tamanho 600x400px.

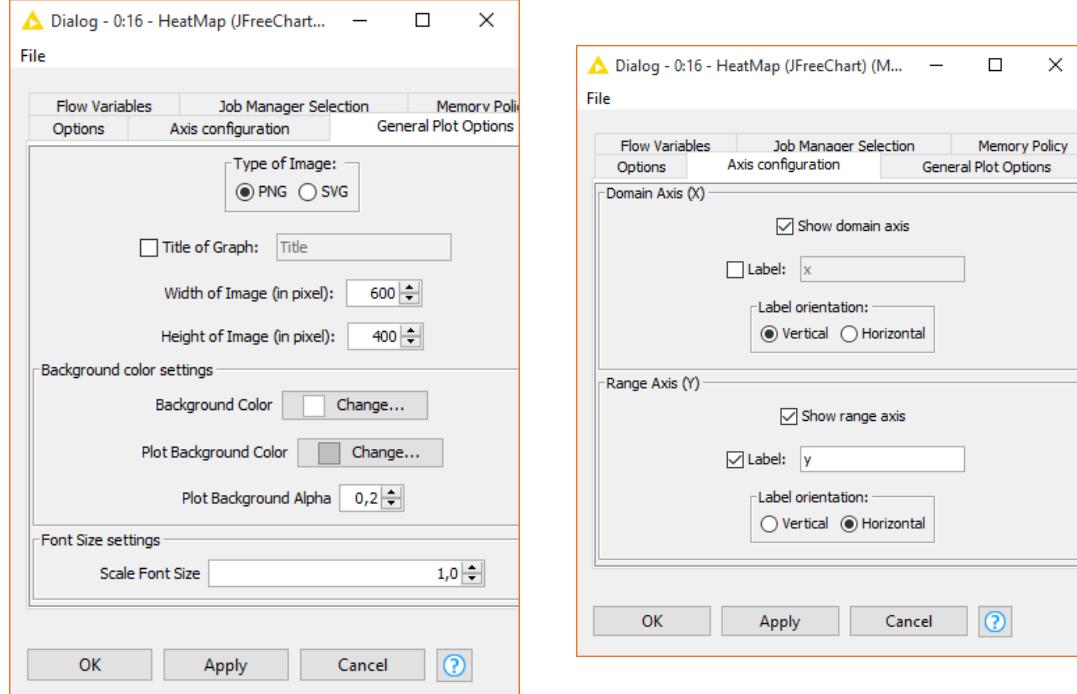


Figura 3.65: Seção de opções de plotagem do nó *HeatMap* no *workflow* de geração da Matriz de coocorrência

Figura 3.66: Seção de configuração de eixos do nó *HeatMap* no *workflow* de geração da Matriz de coocorrência

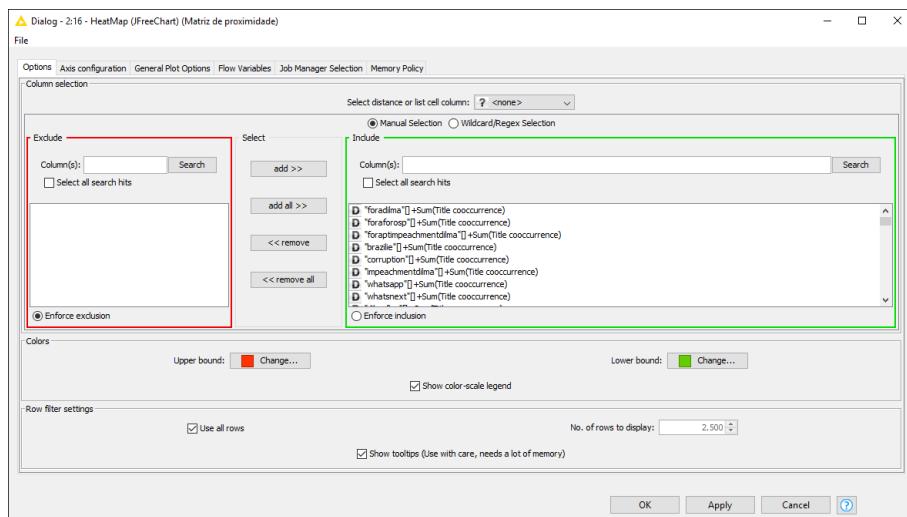


Figura 3.67: Seção de opções da configuração do nó *Heatmap* no *workflow* de geração da Matriz de coocorrência

3.3.7 Aplicação da regra de associação

Nesta subseção é apresentado o *workflow*, apresentado na figura 3.68, que objetiva a visualização das *hashtags* que mais se associam. Para isto, as *hashtags* previamente tratadas são utilizadas para o processamento da regra de associação [62, 60], onde há a utilização do algoritmo *Apriori* [62].

A regra de associação [62, 60] tem o formato $A \rightarrow B$, onde A é chamado de antecedente, e B, de consequente. Pode ser lida como: o atributo(ou item) A frequentemente implica no atributo(ou item) B. Para avaliar as regras geradas são utilizadas algumas medidas, como suporte e confiança.

O suporte é o percentual de vezes que o atributo(ou item) A aparece no conjunto de transações. Nas configurações utilizadas, o valor 30% indica o suporte mínimo necessário para que a regra seja aceita.

A confiança indica o percentual de ocorrência da regra. Nas configurações utilizadas o valor mínimo de confiança utilizado foi 30%.

Os valores de 30% para suporte e 30% para confiança foram obtidos após testes com diversos valores no *workflow*, estes foram considerados bons valores diante da grande quantidade de termos, além da diversidade desses.

O *workflow* abaixo exibe as fases de obtenção dos dados, de preparação e processamento desses dados e da exibição do resultado.

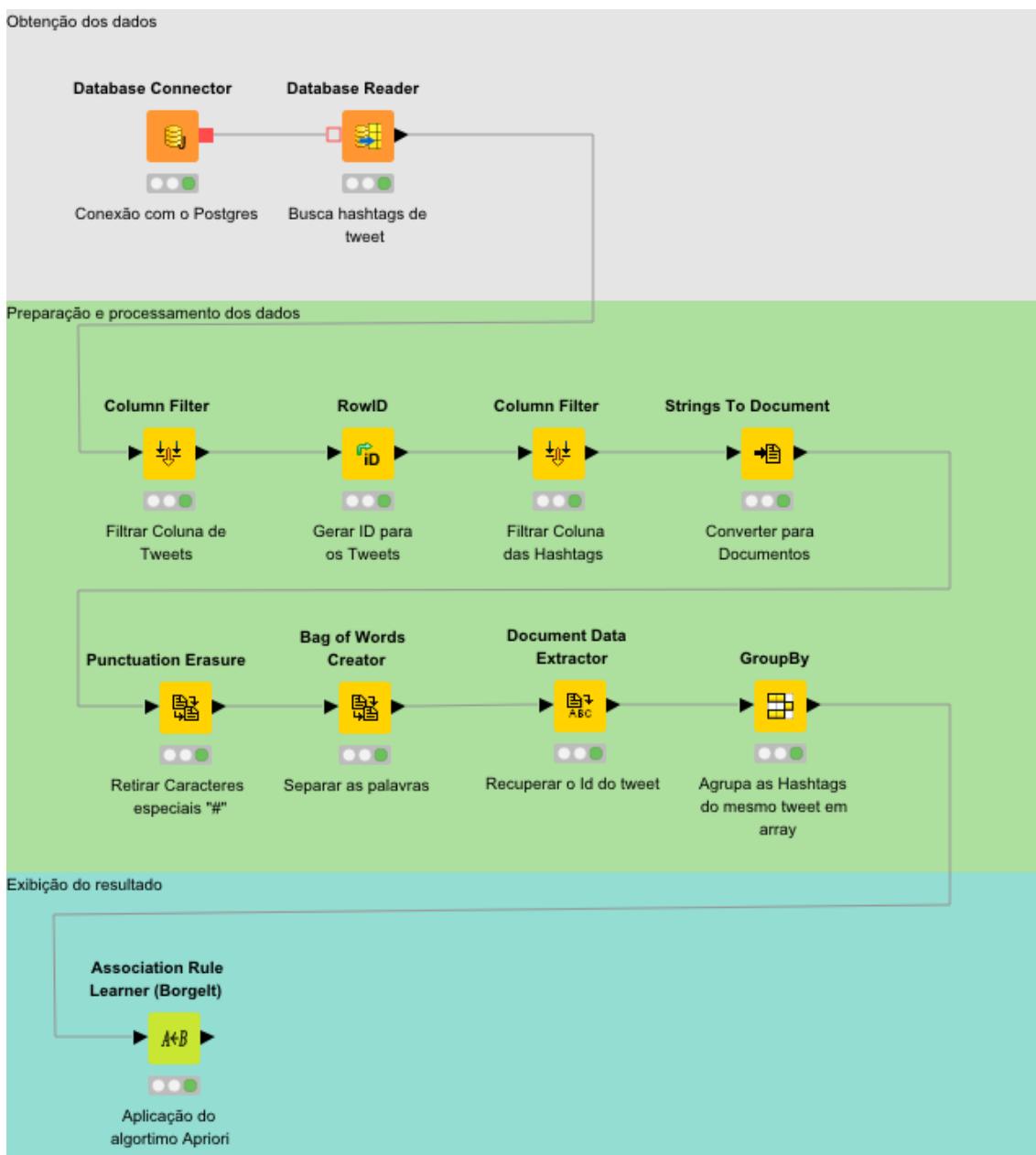


Figura 3.68: Workflow de aplicação da regra de associação

3.3.7.1 Preparação e processamento dos dados

A etapa referente ao tratamento dos dados é exibida na figura 3.69. Os dados obtidos na seção 3.3.2 são formatados e adaptados para utilização no nó de associação.

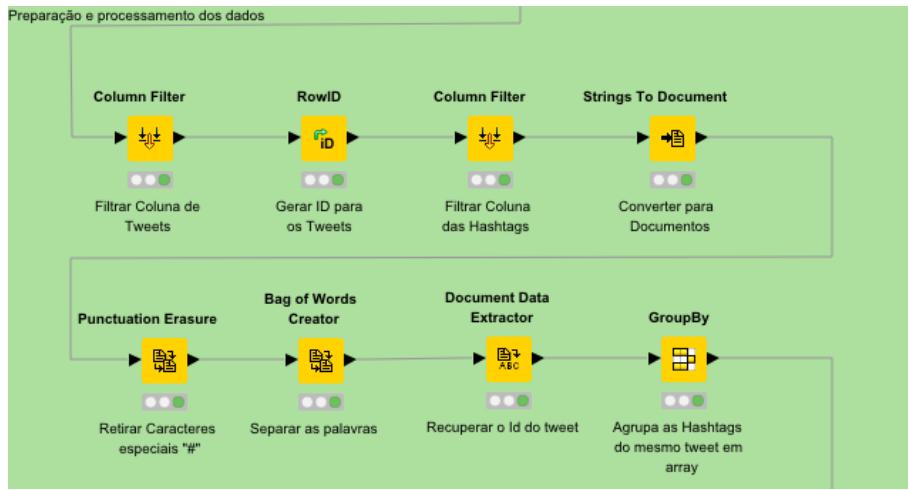


Figura 3.69: Etapa de preparação e processamento dos dados do *workflow* de aplicação da regra de associação

O nó *Column Filter* [6], apresentado na figura 3.70, é responsável pela seleção das colunas que serão utilizadas no próximo nó. No caso, foi selecionada a coluna “*hashtags*”, como pode ser visto na figura 3.71.

O nó *RowID* [33], apresentado na figura 3.72, é responsável por criar um identificador para cada registro obtido do nó anterior. No *workflow* apresentado, foi criado uma nova coluna com a identificação e de nome “*Id_tweet*”, como pode ser visto na figura 3.73.

Após a criação do identificador, o nó *Column Filter* [6], apresentado na figura 3.74, é utilizado possibilitando a seleção das colunas “*hashtags*” e “*Id_tweet*” para a utilização no próximo nó, como pode ser visto na figura 3.75.

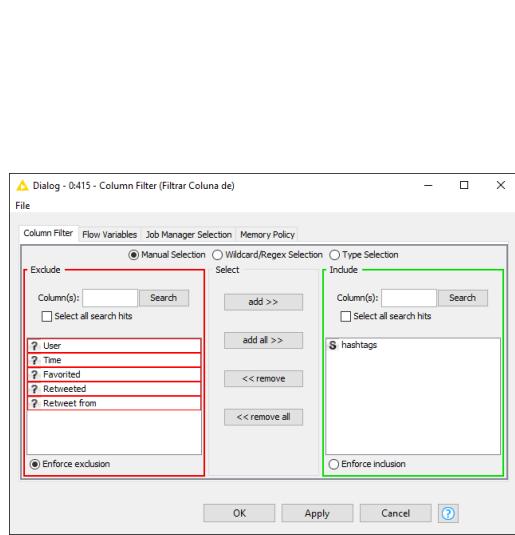


Figura 3.70: Configuração do nó *Column Filter* no *workflow* de aplicação da regra de associação

Properties		Flow Variables
Table "default" - Rows: 107865		Spec - Column: 1
Row ID	hashtags	
Row0	#vemprarua #foradilma	
Row1	#impeachmentdilma #nata... ...	
Row2	#vou13dez #povonarua13d... ...	
Row3	#cambioirreversible #venez... ...	
Row4	#impeachmentdilma #nata... ...	
Row5	#vemprarua13dez #foradilma	
Row6	#foradilma #forapt	
Row7	#cambioirreversible #venez... ...	
Row8	#foramaduro #foradilma #f... ...	
Row9	#cambioirreversible #venez... ...	
Row10	#cambioirreversible #venez... ...	
Row11	#cambioirreversible #venez... ...	
Row12	#impeachmentdilma #vempr... ...	
Row13	#impeachmentja #dilmainco... ...	
Row14	#vemprarua #foradilma	
Row15	#foracunha #cunhanacademia	
Row16	#foradilma #foracunha #va...	

Figura 3.71: Resultado da execução do nó *Column Filter* no *workflow* de aplicação da regra de associação

3.3. ESPECIFICAÇÃO DO WORKFLOW

58

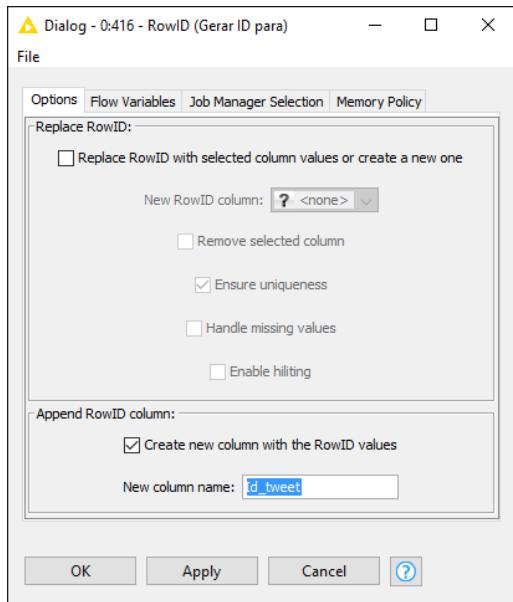


Figura 3.72: Configuração do nó *RowID* no *workflow* de aplicação da regra de associação

Processed data - 0:416 - RowID ...		
File		
Properties		Flow Variables
Table "default" - Rows: 107865		Spec - Columns: 2
Row ID	\$ hashtags	\$ Id_tweet
Row0	#vemprarua #foradilma	Row0
Row1	#impeachmentdilma #natal...	Row1
Row2	#vou13dez #povonarua...	Row2
Row3	#cambioirreversible #ven...	Row3
Row4	#impeachmentdilma #nat...	Row4
Row5	#vemprarua13dez #fora...	Row5
Row6	#foradilma #forapt	Row6
Row7	#cambioirreversible #ven...	Row7
Row8	#foramaduro #foradilma ...	Row8
Row9	#cambioirreversible #ven...	Row9
Row10	#cambioirreversible #ven...	Row10
Row11	#cambioirreversible #ven...	Row11
Row12	#impeachmentdilma #ve...	Row12
Row13	#impeachmentja #dilmaint...	Row13
Row14	#vemprarua #foradilma	Row14
Row15	#foracunha #cunhanacadeia	Row15
Row16	#foradilma #foracunha #...	Row16

Figura 3.73: Resultado da execução do nó *RowID* no *workflow* de aplicação da regra de associação

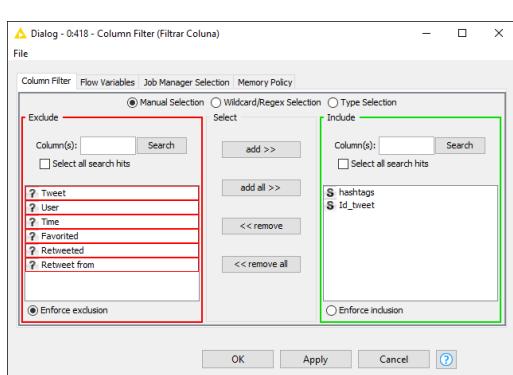


Figura 3.74: Configuração do nó *Column Filter* no *workflow* de aplicação da regra de associação

Filtered table - 0:418 - Column Fi...		
File		
Properties		Flow Variables
Table "default" - Rows: 107865		Spec - Columns: 2
Row ID	\$ hashtags	\$ Id_tweet
Row0	#vemprarua #foradilma	Row0
Row1	#impeachmentdilma #natal...	Row1
Row2	#vou13dez #povonarua13...	Row2
Row3	#cambioirreversible #ven...	Row3
Row4	#impeachmentdilma #natal...	Row4
Row5	#vemprarua13dez #foradilma	Row5
Row6	#foradilma #forapt	Row6
Row7	#cambioirreversible #ven...	Row7
Row8	#foramaduro #foradilma #...	Row8
Row9	#cambioirreversible #ven...	Row9
Row10	#cambioirreversible #ven...	Row10
Row11	#cambioirreversible #ven...	Row11
Row12	#impeachmentdilma #vemp...	Row12
Row13	#impeachmentja #dilmaint...	Row13
Row14	#vemprarua #foradilma	Row14
Row15	#foracunha #cunhanacadeia	Row15
Row16	#foradilma #foracunha #...	Row16

Figura 3.75: Resultado da execução do nó *Column Filter* no *workflow* de aplicação da regra de associação

As etapas de conversão de *strings* para documentos, retirada de cerquilha ("#"), criação de repositório de termos de cada documento, são idênticas as praticadas na seção 3.3.3.1.

O uso do nó *Document Data Extractor* [13], apresentado na figura 3.76, extrai informações de um documento, criando uma nova coluna com essas, no *workflow* foi selecionada as informações da coluna “*Author*”, como pode ser visto na figura 3.77.

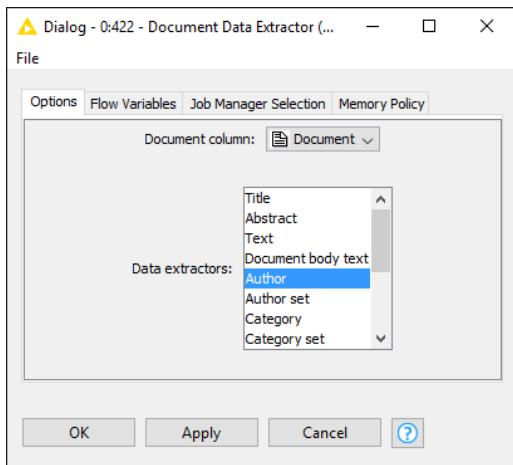


Figura 3.76: Configuração do nó *Document Data Extractor* no *workflow* de aplicação da regra de associação

Row ID	\$ hashtags	\$ Id_tweet
Row0	#vemprarua #foradilma	Row0
Row1	#impeachmentdilma #natal...	Row1
Row2	#vou13dez #povonaruau13...	Row2
Row3	#cambioirreversible #vene...	Row3
Row4	#impeachmentdilma #natal...	Row4
Row5	#vemprarua13dez #foradilma	Row5
Row6	#foradilma #forapt	Row6
Row7	#cambioirreversible #vene...	Row7
Row8	#foramaduro #foradilma #...	Row8
Row9	#cambioirreversible #vene...	Row9
Row10	#cambioirreversible #vene...	Row10
Row11	#cambioirreversible #vene...	Row11
Row12	#impeachmentdilma #vemp...	Row12
Row13	#impeachmentja #dilmaint...	Row13
Row14	#vemprarua #foradilma	Row14
Row15	#foracunha #cunhanacadeia	Row15
Row16	#foradilma #foracunha #v...	Row16

Figura 3.77: Resultado da execução do nó *Document Data Extractor* no *workflow* de aplicação da regra de associação

Após essas etapas, é realizado um agrupamento das *hashtags* de um mesmo *tweet* em um *array* para ser utilizado no nó de associação. O nó que realiza esse agrupamento é o *GroupBy* [14], apresentado na figura 3.78 e com resultado na figura 3.79.

3.3. ESPECIFICAÇÃO DO WORKFLOW

60

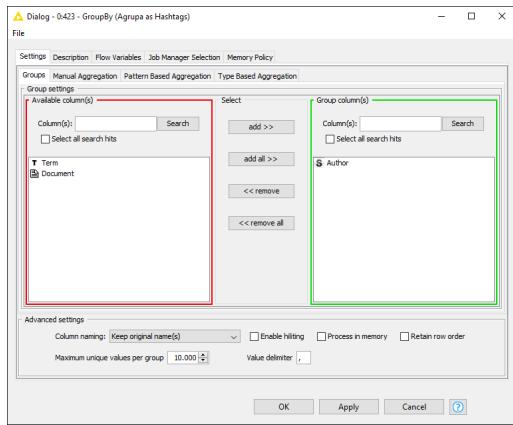


Figura 3.78: Configuração do nó *GroupBy* no *workflow* de aplicação da regra de associação

Table 'default' - Rows: 358797 Spec - Columns: 3 Properties Flow Variables			
Row ID	T Term	Document	S Author
Row1	vemprarua[]	"vemprarua foradima"	- Row0
Row2	foradima[]	"vemprarua foradima"	- Row0
Row3	impeachmen...	"impeachmentdilma nataisemdlma imp...	- Row1
Row4	nataisemdl...	"impeachmentdilma nataisemdlma imp...	- Row1
Row5	impeachmen...	"impeachmentdilma nataisemdlma imp...	- Row1
Row6	povonarua...	"impeachmentdilma nataisemdlma imp...	- Row1
Row7	vemprarua2...	"impeachmentdilma nataisemdlma imp...	- Row1
Row8	sedlmaciob...	"impeachmentdilma nataisemdlma imp...	- Row1
Row9	foradima[]	"impeachmentdilma nataisemdlma imp...	- Row1
Row10	vou13dez[]	"vou13dez povonarua13dez foradima"	- Row2
Row11	povonarua1...	"vou13dez povonarua13dez foradima"	- Row2
Row12	foradima[]	"vou13dez povonarua13dez foradima"	- Row2
Row13	cambioirrev...	"cambioirreversible venezuelalibredem...	- Row3
Row14	venezuelalib...	"cambioirreversible venezuelalibredem...	- Row3
Row15	venezuelalib...	"cambioirreversible venezuelalibredem...	- Row3
Row16	foradima[]	"cambioirreversible venezuelalibredem...	- Row3
Row17	impeachmen...	"impeachmentdilma nataisemdlma imp...	- Row4
Row18	nataisemdl...	"impeachmentdilma nataisemdlma imp...	- Row4

Figura 3.79: Resultado da execução do nó *GroupBy* no *workflow* de aplicação da regra de associação

3.3.7.2 Exibição do Resultado

Nesta subseção, o *workflow* é finalizado processando e exibindo uma lista com as associações obtidas, para tal, o componente *Association Rule Learner(Borgelt)* [2], apresentado na figura 3.80, é utilizado.

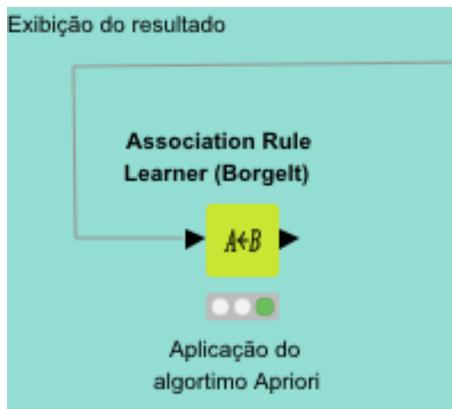


Figura 3.80: Etapa de exibição do resultado no *workflow* de aplicação da regra de associação

Nas configurações, apresentadas na figura 3.81, são selecionadas a coluna com os termos, o valor mínimo de suporte e de confiança, e se é utilizado valor absoluto ou porcentagem para as seleções.

No workflow, foram escolhidas a coluna “Term”, os valores 30% para suporte e confiança e a utilização de porcentagem para seleção.

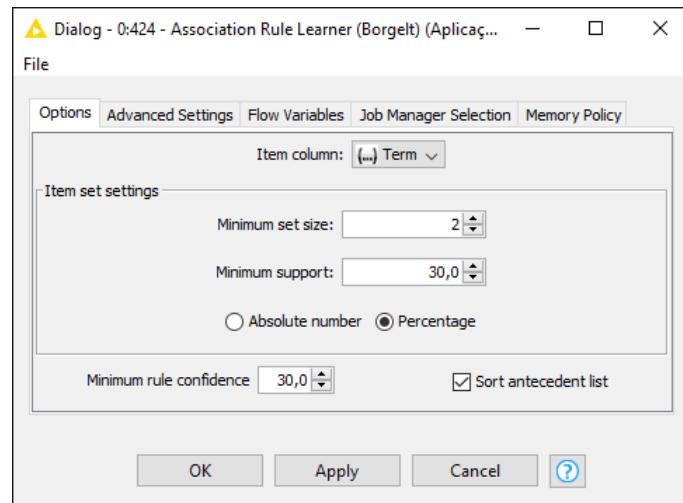


Figura 3.81: Configuração do nó *Association Rule Leaner* no *workflow* de aplicação da regra de associação

Capítulo 4

Resultados e Discussão

4.1 Descrição geral dos experimentos

Para o experimento foi realizada a obtenção dos *tweets* e dados relacionados, como data e hora de criação, quantidade de *retweets*, usuário e suas informações, que possuem as *hashtags* `#foraDilma` ou `#foraCunha`, no período de 20/11/2015 a 31/12/2015. Estas *hashtags* foram escolhidas após uma análise breve por termos mais utilizados para manifestações de rejeição à atos de Dilma e Eduardo Cunha.

Tais *tweets* obtidos são utilizados para extração das *hashtags*, sendo estas tratadas e armazenadas. O conjunto de dados relacionados ao *tweet* e suas *hashtags* são salvos em uma base de dados para utilização em processos posteriores. A descrição do processo de obtenção dos *tweets* é detalhado na seção 3.3.1.

Após a obtenção e tratamento das *hashtags*, foi avaliada a necessidade de tratamento mais sofisticado, onde foi utilizada a distância de Levenshtein [1, 61] para determinar termos muito próximos léxica e semanticamente. Isto possibilita a construção de um dicionário para reparo das *hashtags* na base de dados. A elucidação acerca da elaboração do dicionário de termos é realizada na seção 3.3.4.

Com as *hashtags* reparadas, na seção 3.3.3, um gráfico de frequência foi criado para visualização das *hashtags* mais frequentes nos *tweets*. Tal frequência foi utilizada nos próximos processos para seleção das *hashtags* utilizadas.

Para finalizar os experimentos, a geração da matriz de coocorrência de termos, seção 3.3.5, e geração das regra de associação [62, 60], 3.3.6, são processadas.

4.2 Dados do Twitter

Como apresentado na seção 3.3.1, a extração das informações foi feita por meio da ferramenta KNIME [27], que se conecta diretamente à API do Twitter e possibilita a busca por um termo bem como tweets e dados relacionados ao mesmo.

Com base nos acontecimentos políticos ocorridos foi decidido estudar as hashtags: #foraCunha e #foraDilma. Tais hashtags possibilitaram a aquisição de mais de 100 mil tweets que possuíssem mais de uma hashtag para utilização nos processos de matriz de coocorrência e regra de associação [62, 60].

Após a aquisição, as hashtags foram extraídas dos tweets e armazenadas em um campo separado para facilitar processamentos posteriores. Os dados obtidos são relacionados ao tweet e o usuário que postou.

Nas tabelas a seguir é possível visualizar a estrutura dos dados obtidos. Na tabela 4.1 e 4.2, os campos relacionados ao *tweet* e usuário são exibidos respectivamente. Ambas tabelas demonstram alguns exemplos práticos e reais dos dados.

Tabela 4.1: Estrutura e exemplos de dados do *tweet*

<i>tweet</i>	<i>time</i>	<i>favorited</i>	<i>retweeted</i>	<i>retweet_from</i>
'@guimaraes13PT Grande coisa este cantor ser contra, importante é que o povo brasileiro quer #ForaDilma #ImpeachmenJa'	'2015-12-08 16:48:12'	0	0	
'#RedesSociais: O #ForaCunha embrabalha o jogo do impeachment nas redes sociais - Folha Nobre https://t.co/PzoyJc32e5'	'2015-12-06 12:54:02'	0	0	
'RT @hugobbezerra: Congresso não pode seguir a orientação do chantagista #EduardoCunha: https://t.co/kHCCJZoT6w #ForaCunha #DilmaFica'	'2015-12-05 04:49:43'	0	11	'hugobbezerra'
'Chega dessa máfia !!! #ForaGlobo!!!!#ForaCunha!!!! https://t.co/g0KlCT2628'	'2015-12-16 20:35:12'	0	0	
'RT @PTnaCamara: #NaoVaiTerGolpe: Dilma rebate golpismo de Eduardo Cunha, o das contas não declaradas na Suíça.. #ForaCunha https://t.co/QmA...'	'2015-12-02 20:41:44'	0	31	'PTnaCamara'

Na tabela 4.1 podem ser vistas as seguintes características:

- Campo *tweet* possui o *tweet* postado pelo usuário;
- Campo *time* está relacionado com horário e data da postagem;
- Campo *favorited* exibe a quantidade de curtidas realizadas no *tweet*. A opção curtir, antes chamada de “Favoritar”, possibilita ao usuário demonstrar que gostou de uma publicação sem necessidade de comentário.
- Campo *retweeted* apresenta o número de vezes que o *tweet* foi *retweetado*, onde um usuário divulga uma publicação realizada por outro usuário;
- Campo *retweet_from* exibe o usuário que publicou a publicação *retweetada*.

Tabela 4.2: Estrutura e exemplos de dados do usuário

Campos	Exemplos		
usuario	'Ary_AntiPT'	'guimaraes1985'	'SlimRimografia1'
user_name	'AryAntiPT'	'Guilherme Marques'	'Slim Rimografia (FC)'
user_description	'De uma pá para um pedreiro e ele construirá uma casa! De um livro para uma criança e estará lhe dando o futuro @arykara (A. C. N.)'		'Fã do Slim rapper (@Slim_Rimografia) da melhor qualidade'
user_creation_time	'2014-04-10 19:30:03'	'2012-03-01 20:04:03'	'2014-03-04 02:09:11'
user_language	'pt'	'pt'	'pt'
user_location	'Brasil'		
user_time_zone	'Brasilia'	'Hawaii'	
user_statuses	305881	15080	70546
user_followers	19394	191	1519
user_friends	10352	18	665
user_favourites	68993	52211	13921

Na tabela 4.2 podem ser vistas as seguintes características:

- Campo *usuario* apresenta o *login* utilizado pelo usuário no Twitter;
- Campo *user_name* apresenta o nome registrado pelo usuário;
- Campo *user_description* exibe breve descrição sobre usuário;
- Campo *user_url* está relacionado a um *link* para algum endereço da WEB que o usuário tenha registrado;
- Campo *user_creation_time* apresenta a data e horário de criação do perfil.

- Campo *user_language* exibe a linguagem utilizada;
- Campo *user_location* apresenta a localização atribuída ao usuário;
- Campo *user_time_zone* possui o fuso horário que o perfil se encontra;
- Campo *user_statuses* apresenta a quantidade de *tweets* publicados pelo perfil;
- Campo *user_followers* exibe a quantidade de seguidores do perfil;
- Campo *user_friends* apresenta a quantidade de perfis que o usuário segue;
- Campo *user_favourites* possui a quantidade de *tweets* curtidos pelo usuário.

Os dados pessoais do usuário não foram utilizados na análise, porém foram armazenados para utilização em trabalhos futuros.

O campo *hashtags* não foi apresentado nas tabelas acima por não ter sido obtido diretamente da API do Twitter, e sim através do processo de extração e tratamento, como citado anteriormente. Tal campo tem a estrutura a seguir:

#hashtag1 #hashtag2 #hashtag3 ... #hashtagN

As *hashtags* permanecem com sua trilha (“#”) e separadas por um espaço (“ “). Na tabela 4.3 pode ser visto exemplos de tal estrutura:

Tabela 4.3: Exemplos do campo *hashtags*

Hashtags
"#foradilma #impeachmentja"
"#redessociais #foracunha"
"#eduardocunha #foracunha #dilmafica"
"#dilmarousseff #eduardocunha #foracunha #dilmafica"
"#saude #microcefalia #foradilma #forapt"
"#naovaitergolpe #foracunha"

4.3 Resultados

A figura 4.1, resultado da execução do *workflow* descrito na seção 3.3.4, mostra os valores da frequência das *hashtags* no conjunto de *tweets* obtidos. Pode-se perceber que as *hashtags* utilizadas como base para pesquisa, *#foraDilma* e *#foraCunha*, estão no topo do gráfico e com as maiores frequências, seguidos pela *hashtag* *#foraPT*, em referência ao partido da presidente Dilma Rousseff, *#nãoVaiTerGolpe*, em alusão ao pedido de impeachment realizado por Eduardo Cunha, e *#vemPraRua*, com objetivo de disseminar manifestações.

Pode ser visto no geral *hashtags* em referência ao impeachment da presidente Dilma Rousseff, a favor e contra, menções a manifestações, pedidos de saída de Eduardo Cunha da presidência da Câmara dos Deputados, pedido de prisão aos envolvidos na disputa de poder político, pedidos de saída do presidente do Senado, Renan Calheiros e repúdio a corrupção.

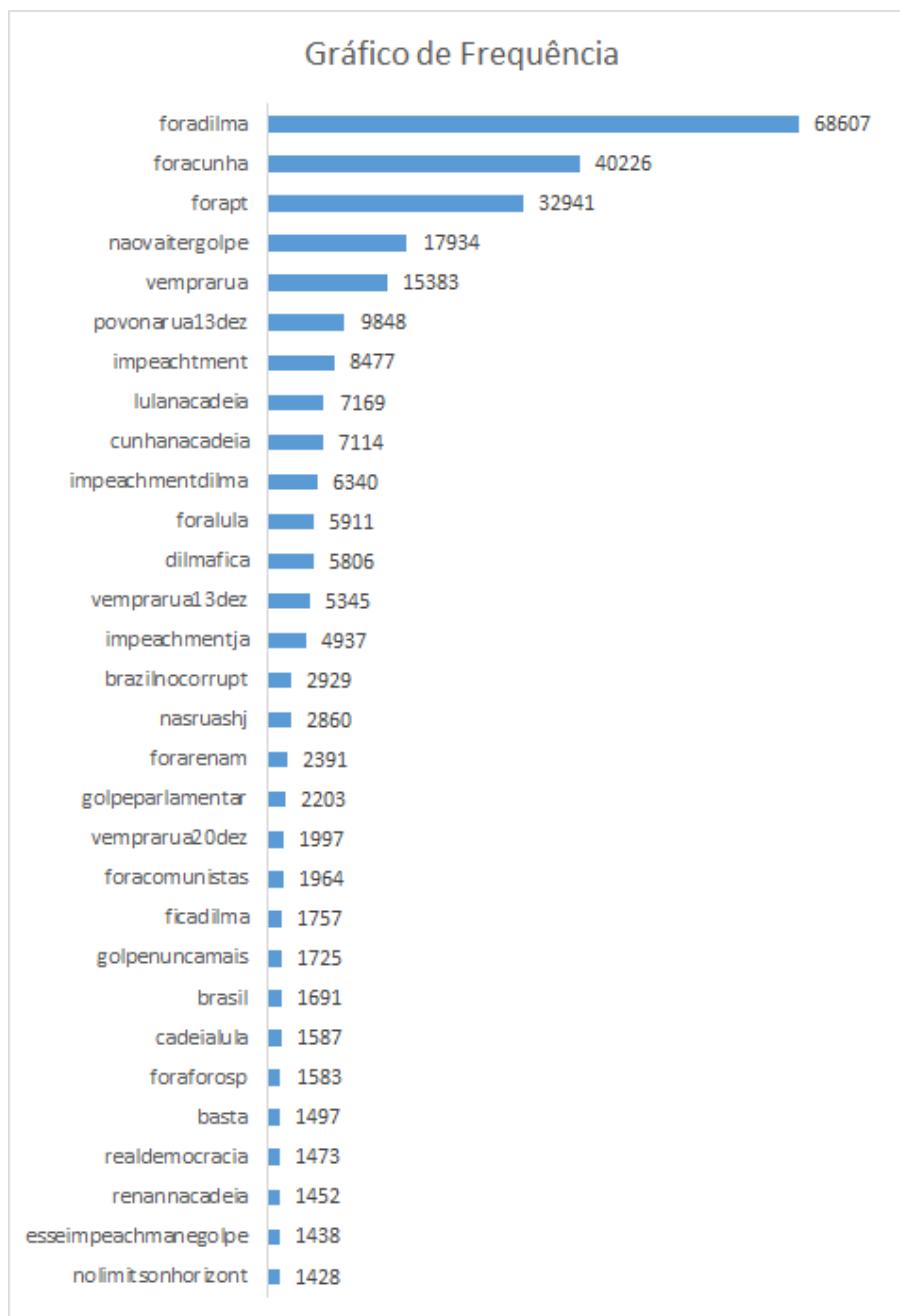


Figura 4.1: Gráfico de frequência

A figura apresentou apenas 2% dos termos mais utilizados na pesquisa, pois seria inviável apresentar todos os termos utilizados nesse tipo de gráfico.

A figura 4.2, obtida através da execução do *workflow* apresentado na seção 3.3.6, exibe os valores de coocorrências entre duas *hashtags* no conjunto de *tweets*. Se destacam os pares de termos: [*foradilma*, *forapt*], em referência ao desejo de retirada

de Dilma Rousseff e seu partido da presidência da república; [*foracunha, naovaiter-golpe*], em alusão a vontade da retirada de Eduardo Cunha da presidência da Câmara dos Deputados e o repúdio ao seu pedido de impeachment contra a presidente Dilma Rousseff. [*foradilma, vemprarua*], novamente em referência ao desejo de retirada de Dilma Rousseff da presidência da república, além de menção a manifestação realizada no dia 13 de dezembro de 2015; e em [*forapt, vemprarua*], há demonstração do desejo pela saída do PT e o incentivo a ocorrências de manifestações.

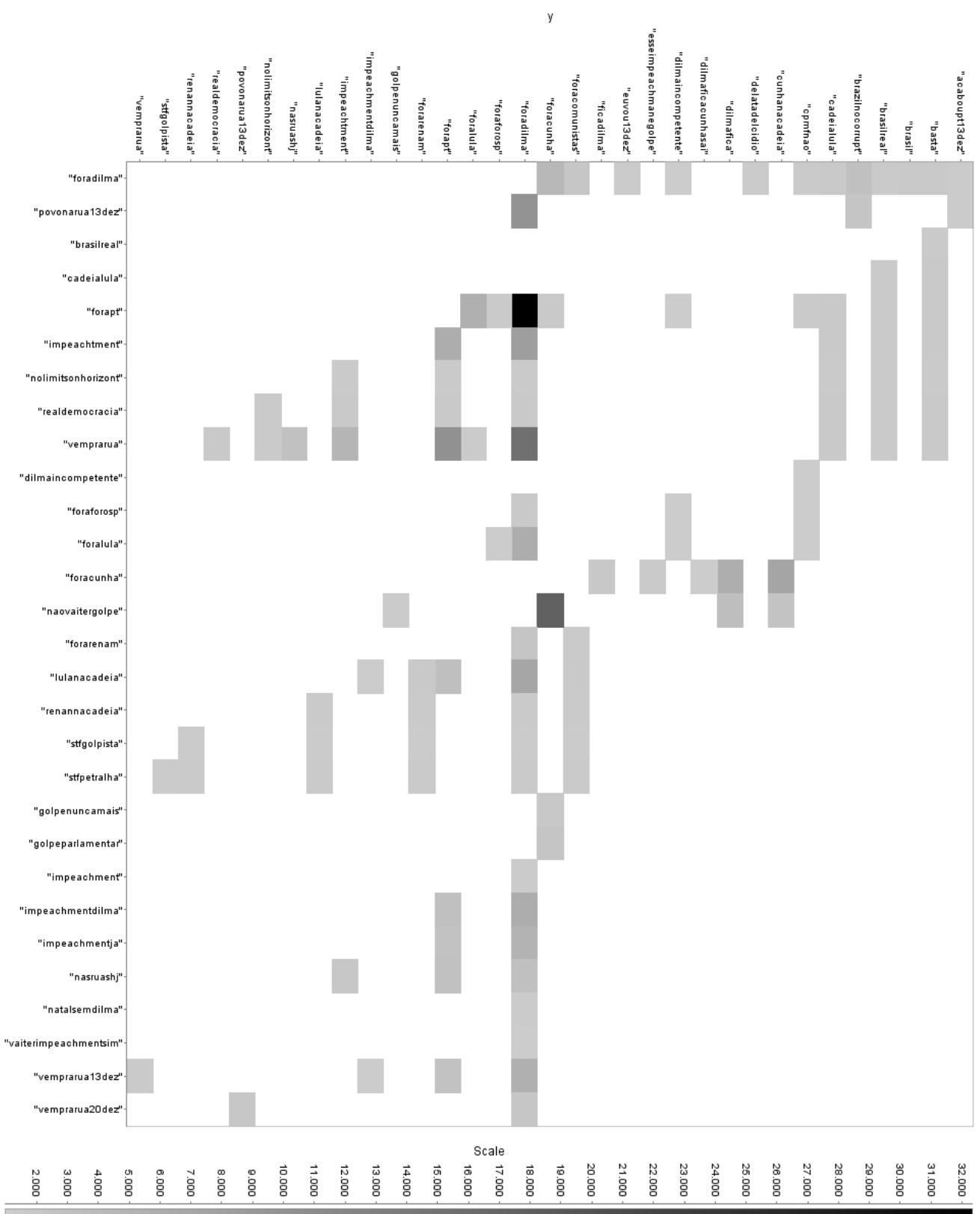


Figura 4.2: Matriz de coocorrência

Na tabela 4.4, obtida através da execução do *workflow* exibido na seção 3.3.7, são apresentadas as regras, através das colunas antecedente e consequente, e valores de confiança e suporte. As regras foram selecionadas por possuírem valores de confiança e suporte acima do mínimo estabelecido de 30%.

A regra tem o formato A->B, onde A é chamado de antecedente e B, de consequente. Pode ser interpretada com a utilização de A frequentemente implica na utilização de B. Um exemplo abaixo é ao usar a *hashtag* `#foraCunha` o usuário tende a usar também a *hashtag* `#nãoVaiTerGolpe`.

A regra (`foradilma -> forapt`) possui suporte de 65.1%, indicando que as *hashtags* `#foradilma` e `#forapt` ocorrem juntas em 65.1% dos *tweets* analisados.

Já a regra (`forapt -> foradilma`) tem confiança de 98.2%, demonstrando que em 98.2% das ocorrências da *hashtag* `#forapt`, ela vem acompanhada de `#foradilma`.

Tabela 4.4: Resultado da aplicação de regra de associação

Antecedente	Consequente	Confiança(%)	Suporte(%)
<code>foracunha</code>	<code>naovaitergolpe</code>	43.5	38.2
<code>forapt</code>	<code>vemprarua</code>	30.9	31.3
<code>forapt</code>	<code>foradilma</code>	98.2	31.3
<code>foradilma</code>	<code>forapt</code>	47.1	65.1
<code>foradilma, forapt</code>	<code>vemprarua</code>	31.0	30.7

4.4 Discussão

Analizando a figura 4.1, é possível observar que a *hashtag* `#foradilma` ocorre aproximadamente 70,7% mais vezes que a `#foracunha`, isso mostra que os usuários do Twitter demonstram ser mais contrários ao governo de Dilma, do que contra as ações de Eduardo Cunha na presidência da Câmara dos Deputados.

Observando também a relação de apoio ou repúdio a Dilma e Eduardo Cunha, pode ser visto que aproximadamente 65% das *hashtags* manifestam-se contra Dilma e

seu partido, o PT, e aproximadamente 30%, contra Eduardo Cunha. O 5% restantes estão relacionados a manifestações de patriotismo, indignação contra corrupção e contra o presidente do Senado, Renan Calheiros.

Após análise, somente da figura 4.1, consegue-se visualizar uma superioridade na frequência de manifestações contra Dilma e seu partido em relação a Eduardo Cunha.

Outra forma possível de análise das manifestações pode ser realizada através da utilização de uma matriz de coocorrências, que pode ser visualizada na figura 4.2. É possível observar neste gráfico que as maiores coocorrências são representadas pela cor mais escura, que neste caso apresenta o par [*#foraDilma, #foraPT*] com maior coocorrência, seguida do par [*#naoVaiTerGolpe, #foraCunha*], par [*#foraDilma, #vemPraRua*], par [*#foraDilma, #povoNaRua13dez*], par [*#cunhaNaCadeia, #foraCunha*] e par [*#dilmaFica, #foraCunha*].

Mesmo com o par [*#foraDilma, #foraPT*] tendo grande destaque diante aos demais, as maiores ocorrências demonstram manifestações contrárias a Dilma e a Eduardo Cunha em um certo grau de equivalência.

O último resultado obtido foi por meio da utilização de Regras de Associação [62, 60] contido na tabela 4.4. Foi possível identificar que a regra de maior suporte, [*foradilma -> forapt*], teve grande destaque diante as demais, com 65,1%, e a regra de maior confiança foi [*forapt -> foradilma*], com 98,2%.

Além dos maiores valores na análise de suporte e confiança, observando todas as regras geradas, grande maioria é contrária a Dilma e seu partido.

Podendo afirmar que através das regras de associação [62, 60], manifestações contra Dilma se destacam em relação a Eduardo Cunha.

Após todas as análises, conclui-se que a partir dos dados coletados do Twitter, no período de dezembro de 2015, pode ser visualizado, em grande destaque, manifestações contra Dilma e seu partido.

Apesar da grande manifestação contrária a Dilma, essa constatação refere-se

apenas ao Twitter, podendo não representar a opinião da maioria da população brasileira.

Capítulo 5

Conclusão e Trabalhos Futuros

Neste trabalho foi criada uma ferramenta para realização de análise e identificação de padrões em *hashtags* no Twitter. Foram criadas etapas de obtenção dos dados, extração e tratamento de *hashtags* para utilização nos processos de mineração e geração de resultados.

Para a mineração foram criadas etapas de confecção de matriz de coocorrências e regras de associação [62, 60] que, após a seleção dos dados de uma lista de frequências de menções a *hashtags*, geraram dados para análise.

Como estudo de caso, foram utilizados *tweets* que continham a *hashtag* `#foraDilma` ou `#foraCunha`, sendo possível encontrar padrões que reforçam que a imagem da presidente Dilma Rousseff está ligada diretamente ao seu partido, o PT, e que a maioria dos registros analisados apresentava repúdio tanto a presidente como ao seu partido. Outro padrão encontrado foi que grande parte das manifestações contrárias a Eduardo Cunha ocorreu em virtude do pedido de Impeachment de Dilma, que para os apoiadores do governo se caracterizou como golpe.

Os resultados mostram a importância da mineração de dados no Twitter para a identificação de padrões comportamentais. Considerando a riqueza de dados que o Twitter possui, a análise realizada neste trabalho pode ser feita utilizando outras *hashtags*, já que esse aspecto, dentro do objetivo proposto, é totalmente configurável.

A seguir, os trabalhos futuros que podem ser gerados:

- Uso de diferentes parâmetros para distância de Levenshtein.
- Pesquisa e utilização de outros tipos de tratamento para as *hashtags*.
- Automatização do processo de seleção de termos para o dicionário.
- Automatização do processo de atualização dos termos do dicionário na base de *hashtags*.
- Automatização da seleção dos dados por frequência de ocorrência
- Pesquisa e testes por diferentes princípios para seleção da frequência de ocorrência de termos em alternativa ao princípio de Pareto.
- Analisar o perfil do usuário que utiliza as *hashtags* alvo de análise, criando estatísticas com relação a tais *hashtags*.
- Aplicação de outras técnicas de mineração de dados

Referências

- [1] Algoritmo de comparação de strings para integração de esquemas de dados.
<http://www.cin.ufpe.br/~tg/2005-2/fmg.pdf>. Acesso em: 23/11/2015.
- [2] Association rule learner (borgelt). https://www.knime.org/files/nodedetails/_mining_subgroup_Association_Rule_Learner_Borgelt_.html. Acesso em: 13/01/2016.
- [3] Bow creator. https://www.knime.org/files/nodedetails/_labs_textprocessing_transformation_BoW_creator.html. Acesso em: 01/10/2015.
- [4] Case converter. https://www.knime.org/files/nodedetails/_labs_textprocessing_preprocessing_Case_converter.html. Acesso em: 17/11/2015.
- [5] Column combiner. https://www.knime.org/files/nodedetails/_manipulation_column_column_split_combine_Column_Combiner.html. Acesso em: 15/12/2015.
- [6] Column filter. https://www.knime.org/files/nodedetails/_manipulation_column_column_filter_Column_Filter.html. Acesso em: 27/11/2015.
- [7] Create collection column. https://www.knime.org/files/nodedetails/_manipulation_column_column_split_combine_Create_Collection_Column.html. Acesso em: 15/12/2015.

- [8] Database connector. https://www.knime.org/files/nodedetails/_database_Database_Connector.html. Acesso em: 01/10/2015.
- [9] Database reader. https://www.knime.org/files/nodedetails/_database_Database_Reader.html. Acesso em: 23/11/2015.
- [10] Database writer. https://www.knime.org/files/nodedetails/_database_Database_Writer.html. Acesso em: 23/11/2015.
- [11] Db drop table. <https://www.knime.org/files/node-documentation/org.knime.base.node.io.database.drop.DBDropTableNodeFactory.html>. Acesso em: 23/11/2015.
- [12] Distance matrix calculate. https://www.knime.org/files/nodedetails/_distmatrix_Distance_Matrix_Calculate.html. Acesso em: 01/12/2015.
- [13] Document data extractor. https://www.knime.org/files/nodedetails/_labs_textprocessing_transformation_Document_Data_Extractor.html. Acesso em: 13/01/2016.
- [14] Groupby. https://www.knime.org/files/nodedetails/_manipulation_row_row_transform_GroupBy.html. Acesso em: 27/11/2015.
- [15] Hashtagify. <http://hashtagify.me/>. Acesso em: 11/10/2015.
- [16] Hashtagify manual. <http://hashtagify.me/manual/api>. Acesso em: 11/10/2015.
- [17] Hashtagify pricing. https://hashtagify.me/pricing?source=presentation_users_lab_page. Acesso em: 11/10/2015.
- [18] Hashtagify princing api. http://hashtagify.me/api_pricing. Acesso em: 11/10/2015.
- [19] Hashtracking. <https://www.hashtracking.com/>. Acesso em: 12/10/2015.
- [20] Hashtracking princing. <https://www.hashtracking.com/pricing/>. Acesso em: 12/10/2015.

- [21] Heatmap (jfreechart). https://www.knime.org/files/nodedetails/_view_jfreechart_HeatMap_JFreeChart_.html. Acesso em: 15/12/2015.
- [22] Insert column header. https://www.knime.org/files/nodedetails/_manipulation_row_row_other_Insert_Column_Header.html. Acesso em: 23/11/2015.
- [23] Java snippet. https://www.knime.org/files/nodedetails/_misc_java_snippet_Java_Snippet.html. Acesso em: 17/11/2015.
- [24] Keyhole. <http://keyhole.co/>. Acesso em: 11/10/2015.
- [25] Keyhole pricing. <http://keyhole.co/pricing>. Acesso em: 11/10/2015.
- [26] Knime. <http://venturess.com/ven-menu-main-services-viz/advanced-analytics/knime>. Acesso em: 20/10/2015.
- [27] Knime about. <https://www.knime.org/about>. Acesso em: 20/10/2015.
- [28] Knime technology partners. <https://www.knime.org/technology-partners>. Acesso em: 20/10/2015.
- [29] Knime twitter nodes. <https://www.knime.org/blog/knime-twitter-nodes>. Acesso em: 15/11/2015.
- [30] O princípio de pareto. <http://www.portaleducacao.com.br/educacao/artigos/26313/o-principio-de-pareto>. Acesso em: 24/12/2015.
- [31] Pair extractor. <https://www.knime.org/files/node-documentation/org.knime.base.node.pair.PairExtractorNodeFactory.html>. Acesso em: 01/12/2015.
- [32] Punctuation erasure. https://www.knime.org/files/nodedetails/_labs_textprocessing_preprocessing_Punctuation_Erasure.html. Acesso em: 27/11/2015.
- [33] Rowid. https://www.knime.org/files/nodedetails/_manipulation_row_row_other_RowID.html. Acesso em: 01/12/2015.

- [34] Significado de diagrama de pareto. <http://www.significados.com.br/diagrama-de-pareto/>. Acesso em: 14/12/2015.
- [35] Significado de workflow. <http://dictionary.cambridge.org/pt/dicionario/ingles/workflow>. Acesso em: 05/11/2015.
- [36] Sorter. https://www.knime.org/files/nodedetails/_manipulation_row_row_transform_Sorter.html. Acesso em: 01/12/2015.
- [37] String distances. <https://www.knime.org/files/node-documentation/org.knime.distance.measure.string.StringDistanceNodeFactory.html>. Acesso em: 01/12/2015.
- [38] String to date time. https://www.knime.org/files/nodedetails/_timeseries_String_to_Date_Time.html. Acesso em: 17/11/2015.
- [39] String to term. https://www.knime.org/files/nodedetails/_labs_textprocessing_transformation_String_to_Term.html. Acesso em: 15/12/2015.
- [40] Strings to document. https://www.knime.org/files/nodedetails/_labs_textprocessing_transformation_Strings_To_Document.html. Acesso em: 23/11/2015.
- [41] Table creator. https://www.knime.org/files/nodedetails/_io_io_other_Table_Creator.html. Acesso em: 23/11/2015.
- [42] Tagboard. <https://tagboard.com/>. Acesso em: 12/10/2015.
- [43] Tagboard solutions. <https://about.tagboard.com/solutions>. Acesso em: 12/10/2015.
- [44] Term co-occurrence counter. https://www.knime.org/files/nodedetails/_labs_textprocessing_frequencies_Term_co_occurrence_counter.html. Acesso em: 15/12/2015.

- [45] Term to string. https://www.knime.org/files/nodedetails/_labs_textprocessing_transformation_Term_to_String.html. Acesso em: 15/12/2015.
- [46] Topsy. <http://topsy.com/>. Acesso em: 11/10/2015.
- [47] Topsy api. <http://api.topsy.com/doc/>. Acesso em: 11/10/2015.
- [48] transform pivoting. https://www.knime.org/files/nodedetails/_manipulation_row_row_transform_Pivoting.html. Acesso em: 15/12/2015.
- [49] Trendsmap. <http://trendsmap.com/>. Acesso em: 12/10/2015.
- [50] Trendsmap pricing. <http://solutions.trendsmap.com/pricing/>. Acesso em: 12/10/2015.
- [51] Tweet binder. <https://www.tweetbinder.com/>. Acesso em: 12/10/2015.
- [52] Tweet binder plans. <https://www.tweetbinder.com/plans>. Acesso em: 12/10/2015.
- [53] Twitter. <https://www.twitter.com>. Acesso em: 30/10/2015.
- [54] Twitter api connector. <https://www.knime.org/files/node-documentation/org.knime.twitter.api.nodes.connector.TwitterAPIConnectorFactory.html>. Acesso em: 15/11/2015.
- [55] Twitter dev. [https://dev.twitter.com/](https://dev.twitter.com). Acesso em: 15/11/2015.
- [56] Twitter followers search. <https://www.knime.org/files/node-documentation/org.knime.twitter.api.nodes.followers.TwitterFollowersSearchFactory.html>. Acesso em: 01/10/2015.
- [57] Twitter search. <https://www.knime.org/files/node-documentation/org.knime.twitter.api.nodes.search.TwitterSearchFactory.html>. Acesso em: 15/11/2015.

- [58] Value counter. https://www.knime.org/files/nodedetails/_statistics_Value_Counter.html. Acesso em: 01/12/2015.
- [59] Workflow based framework for life science informatics. <http://www.sciencedirect.com/science/article/pii/S1476927107001107>. Acesso em: 20/10/2015.
- [60] HAN, J., E KAMBER, M. Data mining: Concepts and techniques, 2000.
- [61] MANNING, CHRISTOPHER D., P. R., E SCHÜTZE, H. Introduction to information retrieval, 2008.
- [62] TAN, P., STEINBACH, M., E KUMAR, V. Introduction to data mining, 2006.